MeV

Modeling • Experimentation • Validation

SUMMER SCHOOL

www.MeVSchool.org

CFD and V&V for CFD

William J. Rider

wjrider@sandia.gov

Computational Multiphysics

Sandia National Laboratories, Albuquerque

Albuquerque, NM 87185

July 23, 2010

SAND2009-4025C

Session #

Date

Time

Who Am I ?

- I'm a staff member at Sandia, and I've been at SNL for 3 1/2 years. Prior to that I was at LANL for 18 years. I've worked in ASC since its beginning and in the ASC V&V program since it began.
- In addition, I have expertise in hydrodynamics (incompressible to shock), numerical analysis, interface tracking, turbulence modeling, nonlinear coupled physics modeling, nuclear engineering…
- I've written two books and lots of papers on these, and other topics.

Acknowledgements

- The MeV school
- Tim Trucano, Marty Pilch, Allen Robinson, Erik, Strack, Greg Weirs, John Niederhaus, Randy Summers, Tony Guinta, John Shadid, and Pavel Bohev (SNL)
- Jim Kamm, Jerry Brock, Fernando Grinstein, Len Margolin, Marv Alme and Bob Webster (LANL)
- Jeff Greenough and Jeff Banks (LLNL)
- Bill Oberkamf (SNL retired)
- Chris Roy (Virginia Tech.)

References for these Lectures

- Richtmyer & Morton, “Difference Methods for Initial Value Problems,” Wiley Interscience, 1967.
- R. J. Leveque, “Nonlinear Conservation Laws,” Birkauser, or his more recent Cambridge Book. 1990.
- Stephen Pope’s Book, Turbulent Flow, Cambridge University Press, 2000.
- Uriel Frisch’s, “Turbulence,” Cambridge University Press, 1995.
- Oberkampf & Trucano, “Verification and Validation in Computational Fluid Dynamics,” Progress in Aerospace Sciences, 2002.
- Pat Roache’s paper in Annual Reviews in Fluid Mechanics, 1998.
- Pat Roache’s book, “*Verification and Validation in Computational Science and Engineering*” – new edition available

References


Lectures in Mathematics ETH Zürich
Randall J. LeVeque
Numerical Methods for Conservation Laws
Birkhäuser
TURBULENCE
Uriel Frisch
Turbulent Flows
Stephen B. Pope
VERIFICATION AND VALIDATION IN COMPUTATIONAL SCIENCE AND ENGINEERING
PATRICK J. ROACHE
FUNDAMENTALS OF
RIFICATION AND VALIDATION
PATRICK J. ROACHE
CAMBRIDGE TEXTS IN APPLIED MATHEMATICS
Finite Volume Methods for Hyperbolic Problems
RANDALL J. LEVEQUE
William L. Oberkampf and Christopher J. Roy
Verification and Validation in Scientific Computing
CAMBRIDGE
Numerical Simulation of Reactive Flow
Second Edition
ELAINE S. ORAN
JAY P. BORIS
Dale R. Durran
NUMERICAL METHODS FOR WAVE EQUATIONS IN GEOPHYSICAL FLUID DYNAMICS
Computational Gasdynamics
Culbert B. Laney

PART 1. BASICS OF CFD

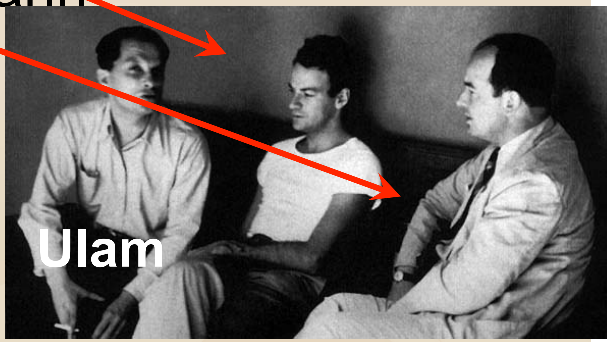
A thought to start us off.

"An expert is someone who knows some of the worst mistakes that can be made in his (her) subject, and how to avoid them."

- Werner Heisenberg

The origin of CFD calculations

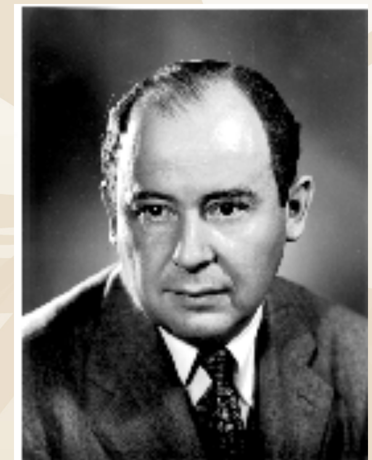
- The first CFD calculation was reported in a Los Alamos report on June 20, 1944 – lead author Hans Bethe
  - Feynmann was the computational lead
  - Still classified!
- The first codes were 1-D and Lagrangian, shocks were tracked (no viscosity & finite differences failed completely as of 1945 w/o artificial viscosity).
- Artificial viscosity was developed by Von Neumann and Richtmyer (Richtmyer published a report in 1948 and is the likely inventor of the concept).

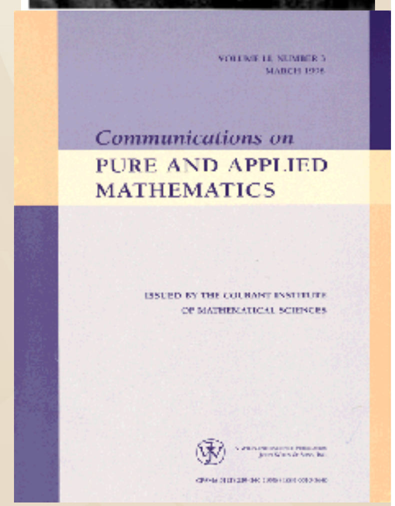


CFD codes were natural for some problems.

- Outside the weapons' labs codes (probably) got their start in numerical weather prediction (via John Von Neumann at IAS)
- At the Weapons Labs – particle-in-cell codes appeared in the mid-1950's at LASL
- Grid based Fluid codes – Noh's CEL code in Methods in Computational Physics, Volume 3 1964
- In the mid-1960's codes began to extend to a wider set of communities such as aerospace, mechanical/ nuclear engineering and astrophysics.
  - Popularized by Harlow (LASL) & Spalding (Imper. College)
- Maturity* in the codes started to be seen by the mid-1970's.

*** Maturity is use beyond "research" codes (i.e. programmatic work), and also depends upon widespread availability of capable computing.**





Reading material: Harlow on the history of fluid dynamics in T-3

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT

Journal of Computational Physics 195 (2004) 414–433

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Review

Fluid dynamics in Group T-3 Los Alamos National Laboratory ☆ (LA-UR-03-3852)

Francis H. Harlow *

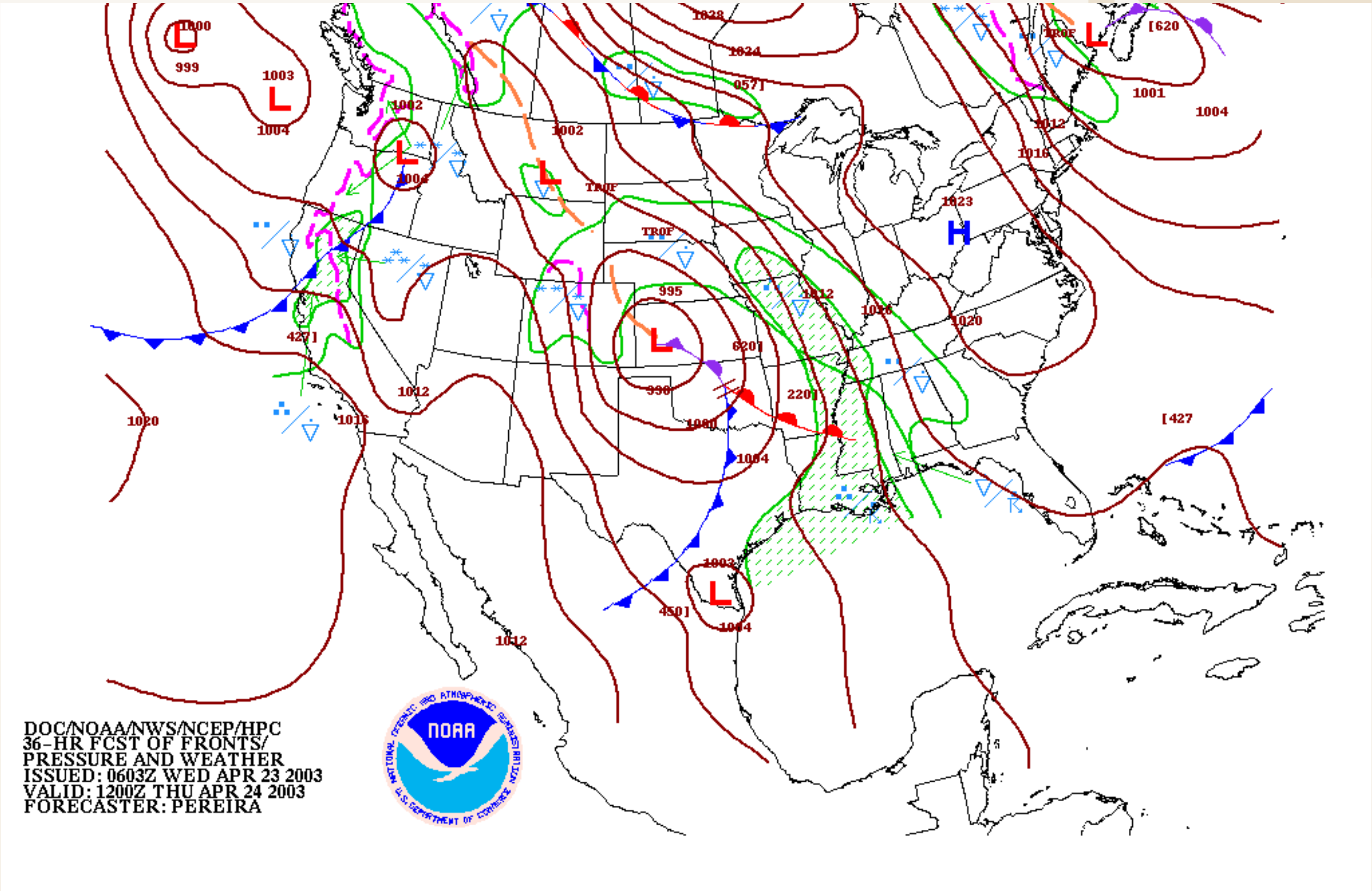
Los Alamos National Laboratory, MS-B216, Los Alamos, NM 87544, USA

Received 12 June 2003; received in revised form 25 August 2003; accepted 3 September 2003

Abstract

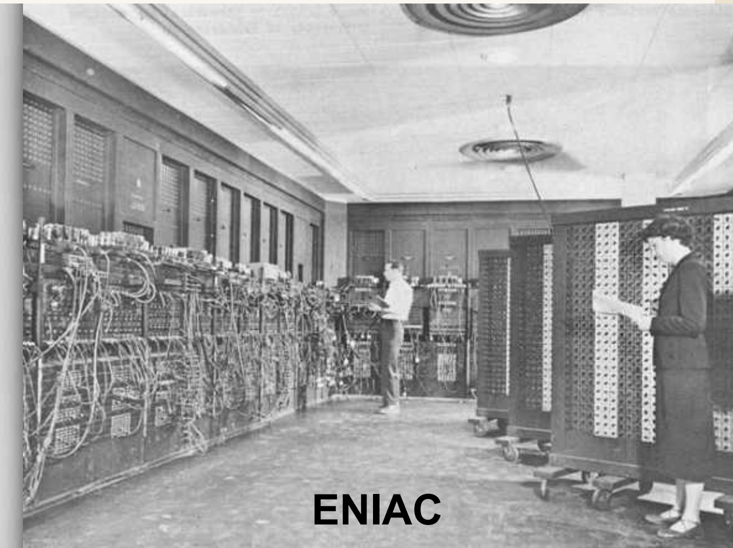
The development of computer fluid dynamics has been closely associated with the evolution of large high-speed computers. At first the principal incentive was to produce numerical techniques for solving problems related to national

The first "real" CFD calculation was for weather forecasting.

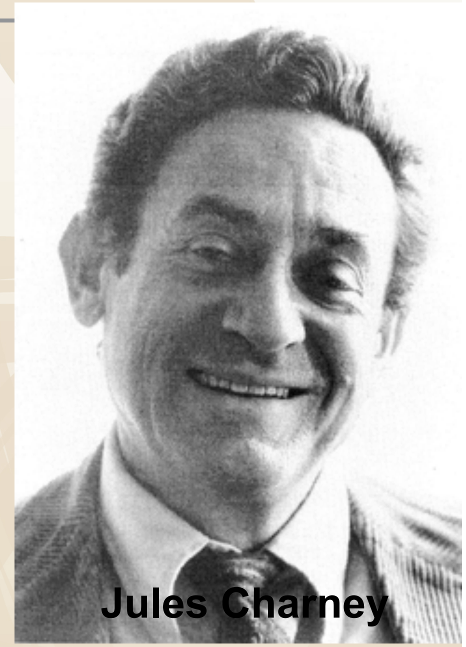


It took place at IAS (Princeton) in 1950 involving, among others, John Von Neumann.

In front of the Eniac, Aberdeen Proving Ground, April 4, 1950, on the occasion of the first numerical weather computations carried out with the aid of a high-speed computer. Left to right: H. Wexler, J. von Neumann, M. H. Frankel, J. Namias, J. C. Freeman, R. Fjortoft, F. W. Reichelderfer, and J. G. Charney.

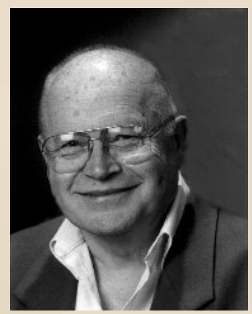


ENIAC

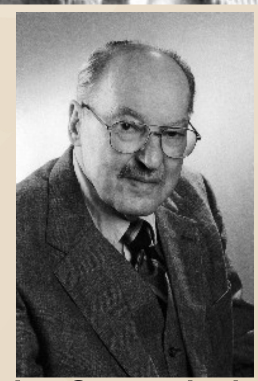


Jules Charney

First calculation
16x16x(3) mesh
Δx=300 km
48 time steps
Δt=30minutes



Norm Phillips



Joe Smagorinsky

Staggered Grid

k	ϕ, ψ, u, v	ϕ, ψ, ρ, u, v	ρ, u, v	$p \downarrow$
$k+½$	ρ, ω	ω	ϕ, ψ, ω	
	(a)	(b)	(c)	

A connection between weather modeling, Von Neumann and Large Eddy Simulation

- In 1956 a simulation by Norm Phillips of weather over the eastern half of the US for a month was completed and the subject of a meeting at IAS.
- Late in the simulation the solution began to experience and instability (ringing)
- It was suggested by Charney that “Von Neumann’s viscosity” might control this ringing.
- Smagorinsky completed the follow on simulation including this technique, which was 3-D instead of 2-D.
  - This technique became the first Large Eddy Simulation (LES) *subgrid* turbulence model
  - *One should note with some significance that the first shock capturing method gave birth to a turbulence model!*

Types of CFD solver: hyperbolic, elliptic and parabolic PDEs

- The starting point for methods is usually a hyperbolic system of PDEs.
  - Methods are often explicit and have a severe time step constraint.
  - Viscous terms are parabolic.
- Incompressible flow involves an elliptic PDE along with both hyperbolic terms, and parabolic viscous terms.
  - The (stability-based) time step is determined by explicit terms (accuracy is a more subtle issue!).
- Many methods utilize (semi-)implicit methods to remove time step restrictions.

Each type of PDE brings substantial, but different numerical challenges.

$$\boxed{\nabla \cdot u = 0} \quad \rightarrow \underbrace{\nabla^2 p = -\nabla \cdot \left(u \cdot \nabla u - \nu \nabla^2 u \right)}_{\text{elliptic}}$$

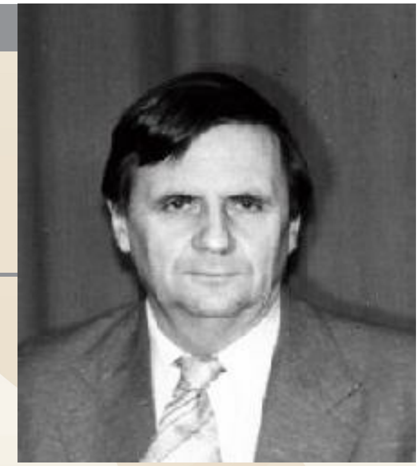
$$\frac{\partial u}{\partial t} + \underbrace{u \cdot \nabla u}_{\text{hyperbolic}} + \boxed{\nabla p} = \underbrace{\nu \nabla^2 u}_{\text{parabolic}}$$

- **Hyperbolic PDEs can support spontaneously developing discontinuous solutions.**
- **Explicit methods for hyperbolic or parabolic PDEs can carry restrictive stability conditions.**
- **Implicit methods for hyperbolic PDEs are expensive and often lack robustness.**
- **Elliptic PDEs are expensive to solve, but generally robust.**
- **Parabolic PDEs are generally easier to solve.**

There are a lot of different numerical methods, but they all depend on the same fundamentals.

- Methods fall into a variety of categories: finite difference, finite volume, finite element, discontinuous Galerkin, spectral, spectral element, spectral volume, semi-Lagrangian, balance etc,…
- For time dependent methods there are explicit, semi-implicit, implicit, linearized, nonlinearly consistent,…
- Different methods are advantageous for different circumstances, applications and other considerations.
- All methods have the same objective solve the governing equations in an accurate, stable and efficient manner,
- They ultimately have to abide by the same fundamental requirements.
- *I will mostly focus on finite difference/volume methods for hyperbolic PDEs because it is what I know the most about!*

Godunov's Theorem relating high-order and montonicity



- Godunov's theorem says that a high-order linear methods (2nd or higher) cannot be monotone for advection.
- Restated: only 1st order linear methods are monotone
- A linear method uses the same differencing stencil for all zones.

- *Godunov also developed a method that has been used extensively in aerospace and astrophysics.*

Godunov’s account of the creation of his method and theorem.

Journal of Computational Physics 153, 6–25 (1999)

SPECIAL ARTICLE

Reminiscences about Difference Schemes[1]

Sergei Konstantinovich Godunov

Stekhlov Institute of Mathematics, Russian Academy of Sciences, Siberian Branch 630090 Novosibirsk, Russia

Received August 31, 1998

FOREWORD

In these notes, I will tell how in 1953–1954 the first version of the “Godunov’s scheme” was invented and how it was modified in the subsequent works by myself (until 1969) and by others at the Institute of Applied Mathematics in Moscow (now named after its founder, academician M. V. Keldish).

Robust methods for hyperbolic PDEs were too dissipative until the early 1970's when…

- … a revolution began in computational physics.
- Within the span of one or two years four researchers independently developed a key idea that made Eulerian codes viable.
- It revolved around solving $$\frac{\partial \varphi}{\partial t} + \nabla \cdot \left(\vec{u} - \vec{u}_{\text{grid}} \right) \varphi = 0$$
- All four developed "high-resolution" methods:
  - Boris (NRL) Flux Corrected Transport
  - Van Leer (Leiden, Netherlands) limiters
  - Kolgan (USSR, Taiga) high-order Godunov
  - Harten (NYU/Israel) self-adjusting hybrid

What the heck was going on in 1971 and 1972? Was something in the water?

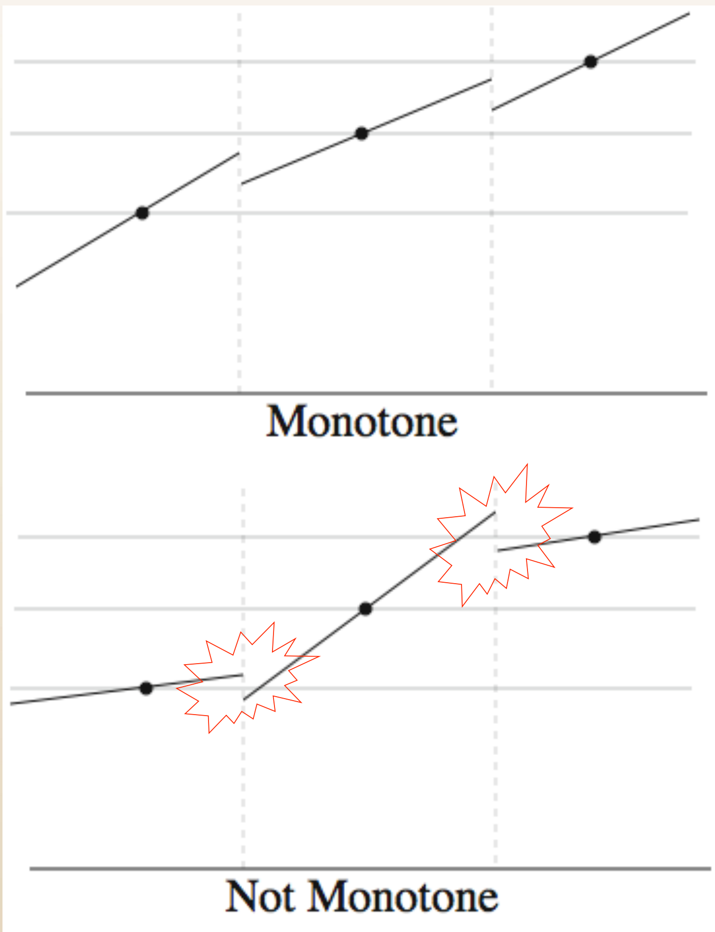
- Eulerian hydrocodes became mainstream with the publication of the Journal of Computational Physics, …
- … and the availability of large scale computing like the CDC machines.
- The Lax-Wendroff method was the mainstay of computations <u>outside</u> the Lab and provided the basis to work from.
- Aerodynamics, combustion and astrophysics communities all started computing (more naturally Eulerian in how the problems were cast).

Overcoming Godunov's Theorem with nonlinear methods for advection

- The key to overcoming Godunov's theorem is using nonlinear methods – using different stencils dependent on the local solution.
- Developed *independently* by four men in 1971-1972
  - Jay Boris (NRL)
  - Bram Van Leer (U. Leiden)
  - Kolgan (USSR)
  - Harten (Israel)

Ami Harten *1947-1994*
The developer of TVD & ENO schemes. We'll always miss your originality in science and in life.
SO, (2003)

Monotonicity is a desirable property to maintain numerically leading to the suppression of oscillations without too much dissipation for advection.



Some of the first high-resolution methods invoked a geometric definition of monotonicity.

For an interpolation the reconstruction of a function in a cell should not exceed the values of its neighboring cells.

Development of implicit methods

- Had the twin genesis of the MAC method (1st) and Chorin's projection (2nd) for incompressible flows.
- The MAC method formed the basis of most developments for incompressible flow solvers used by the engineering community.
- At LASL Harlow and Company started to develop the ICE methods (semi-implicit) which formed the basis of reactive flow and multimaterial codes. These codes used the experience of the MAC method.
- These codes became the basis of early reactor safety codes (TRAC, RELAP, TRACE)
  - Harlow ended research in CFD by ~1975 and began working on turbulence modeling.

High-Order (Projection-like) Methods for Incompressible Flows

- Hyperbolic terms differenced using a variety of high resolution methods (finite volume, Godunov, ENO, FCT, SMART, QUICK-type, etc…)
- Time integration should be done without spatial splitting
- Exact and approximate versions of Chorin's projection method

$$\nabla \cdot \frac{1}{\rho}\nabla p = \nabla \cdot \vec{u}^{*} \quad \Longrightarrow \quad \vec{u}^{n+1} = \vec{u}^{*} - \frac{1}{\rho}\nabla p$$

- Other similar methods are SIMPLE, SIMPLER,… sequential methods with "fixed point" correction.
- More recently methods have become implicit and more fully coupled (like Newton-Krylov).
- Finite element method have under-gone similar developments (see Gresho among others).

Two Step Solution

- 1st solve a convection-diffusion equation

- Plus other equations (density) without being constrained to be divergence free

$$\vec{u}^* = \vec{u}^n - \Delta t\left[\vec{u} \cdot \nabla\vec{u} + \nu\nabla^2\vec{u} + \vec{F}\right]^{n+1/2}$$

- The apply the projection of the velocities to a (approximately) divergence-free subspace

$$\rho^{n+1} = \rho^n - \Delta t \nabla \cdot (\rho\vec{u})^{n+1/2}$$

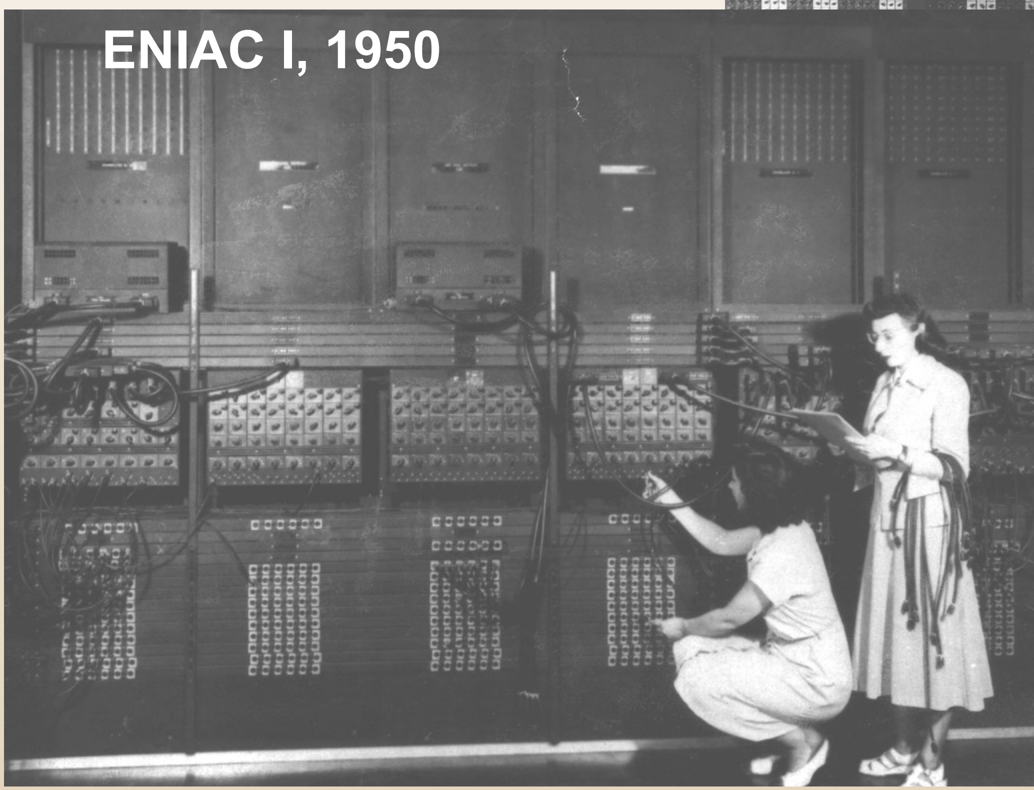
The (approximate) projection & semi-implicit methods are closely related.

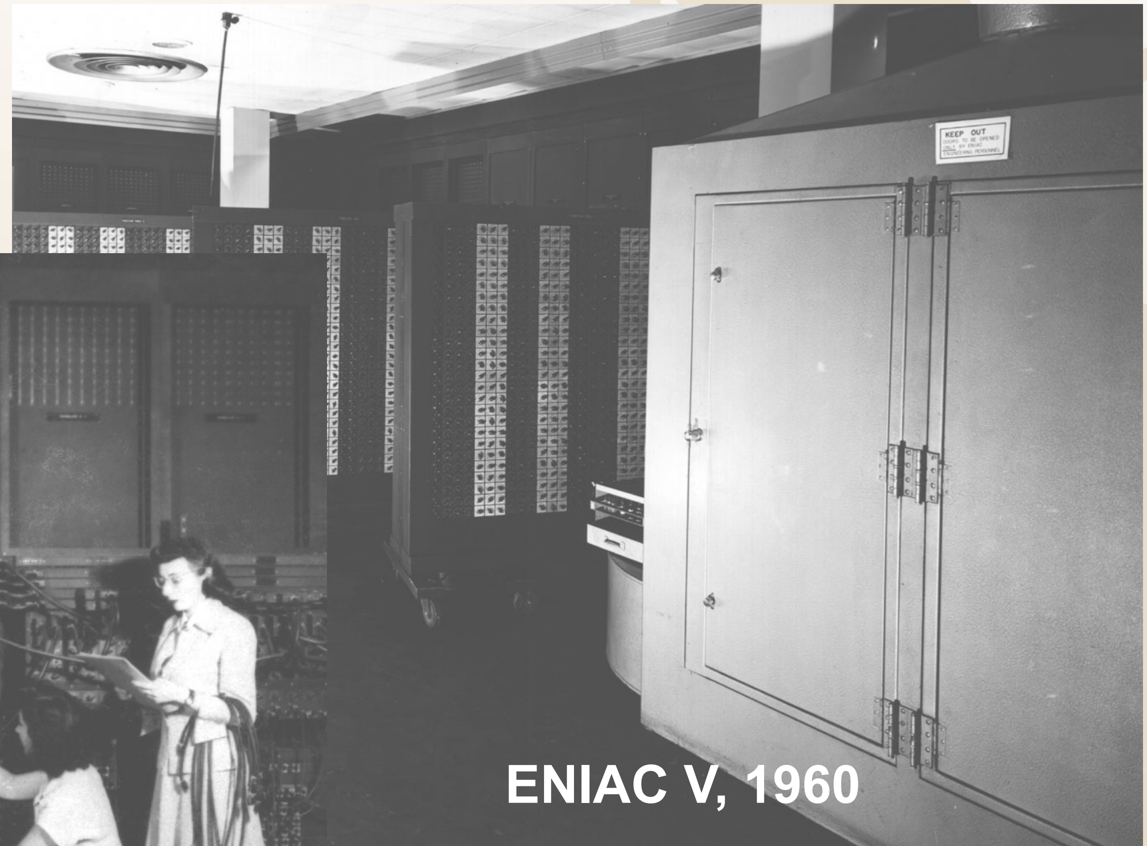
- The pressure Poisson equation
- Solved twice a time step
- Solved using a multigrid preconditioned conjugate gradient method
- The end of time step divergence is only approximately zero

$$\vec{u}^{n+1} = \vec{u}^* - \frac{1}{\rho}\nabla p \quad \Longrightarrow \quad \nabla \cdot \frac{1}{\rho}\nabla p = \nabla \cdot \vec{u}^* \qquad \nabla \cdot \vec{u} \underset{\approx}{=} 0$$

- Semi-implicit (ICE) methods work similarly, except there is no divergence constraint although the elliptic equation is formed in the same way.
  - The implicit step removes the stability restriction associated with sound waves.

The evolution of computers is hard to separate from the history of codes

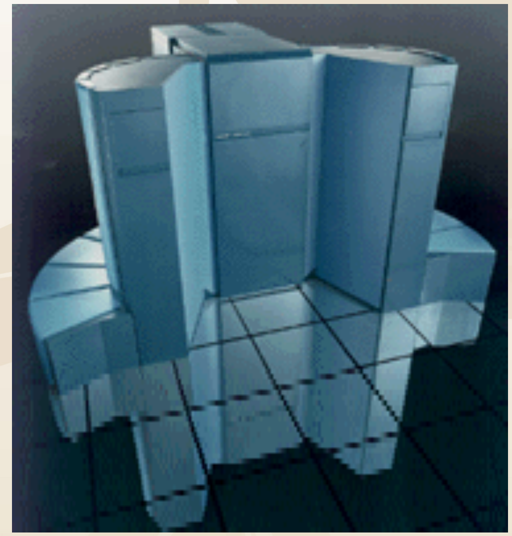


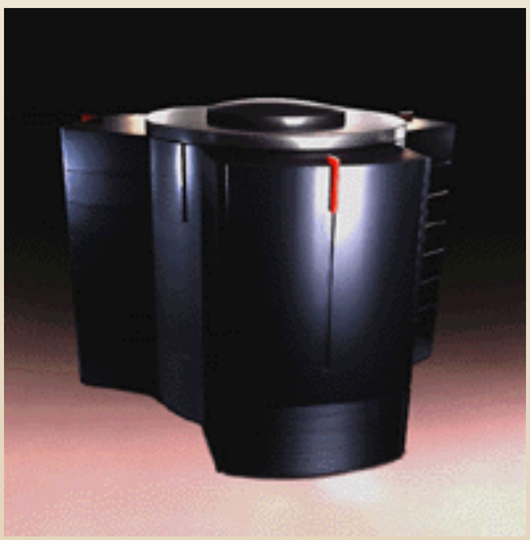


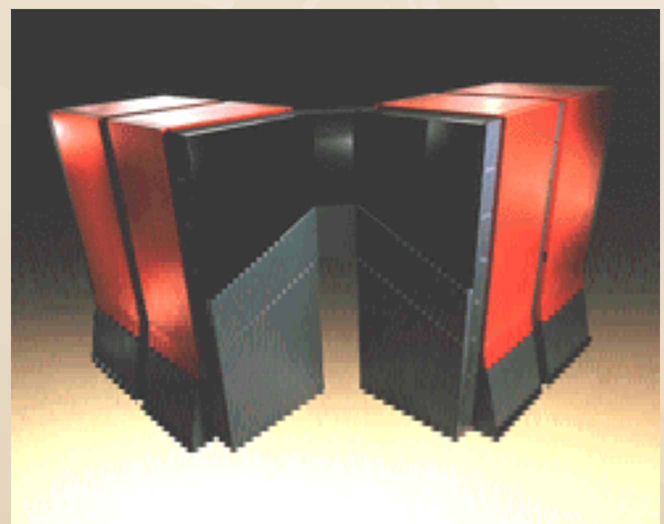
The appearance of the CDC 6600 is an important watershed (mid 60's).

- The combination of new machines and new methods make useful engineering codes possible.

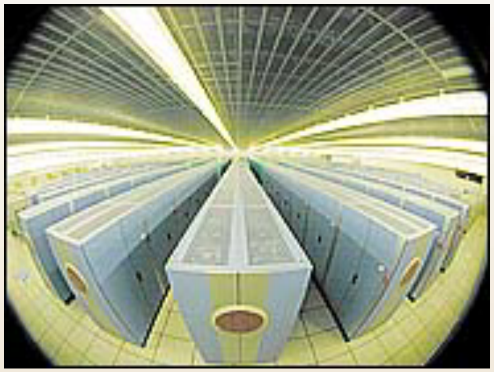
Of course there are Crays from the 70's-90's



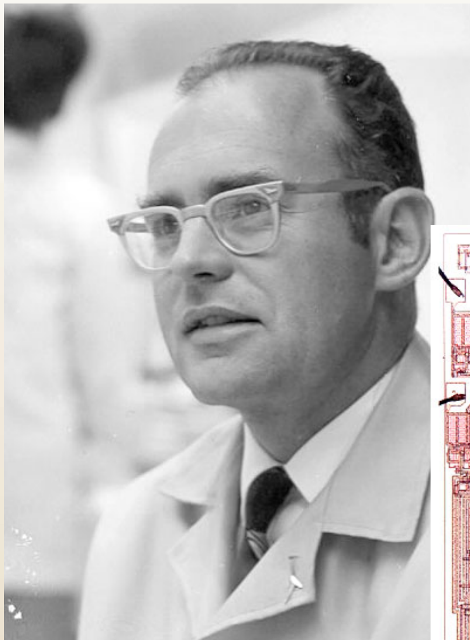




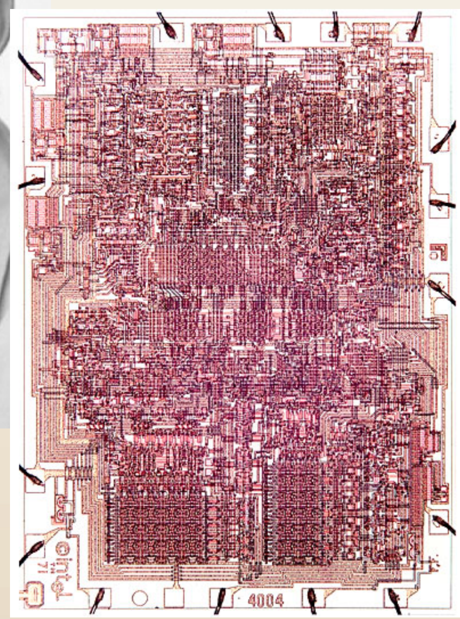
…and the modern ASCI era with room filling machines again!

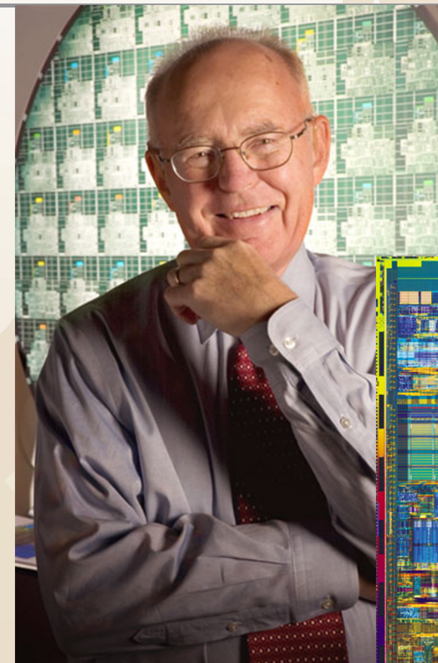


We’ve been gliding along with Moore’s law for 40+ years, will it continue?



Then



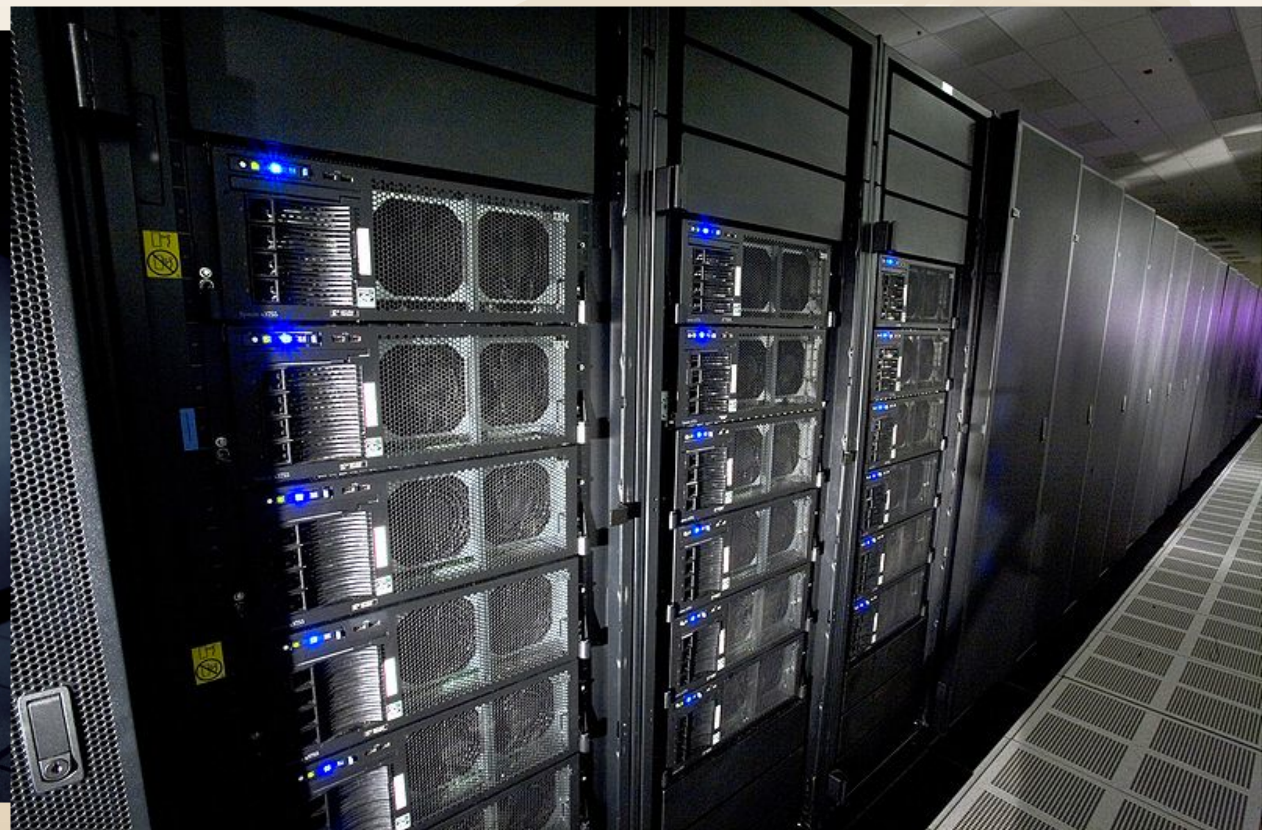


Now

- Recently, IBM, Sony & Toshiba has started putting parallel processors on a chip. Its called Cell (9 proc.)

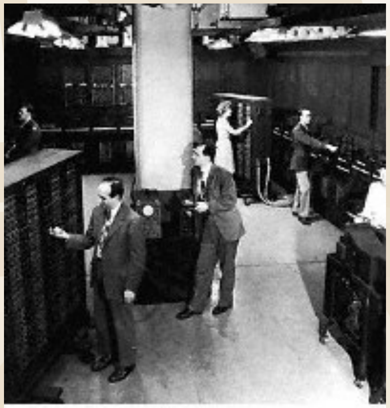
The next generation of computing will be disruptive!

- The programming model developed over the last 15 years will probably be overthrown.
- The consequences for methods is not known!



One important aspect is the development of operating systems and languages

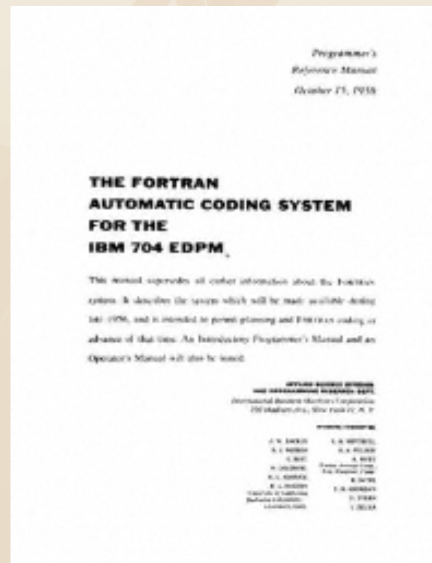
- In the early days of computers programming was much more challenging, even involving the physical modification of the computer in order to implement programs.
- This difficulty limited the complexity of algorithms that one would place on a machine.



One important aspect is the development of operating systems and languages

- The arrival of Fortran was a big advance.
- The next big event was the placement of programs in memory.
- The operating systems were constantly changing, and still do to some extent.

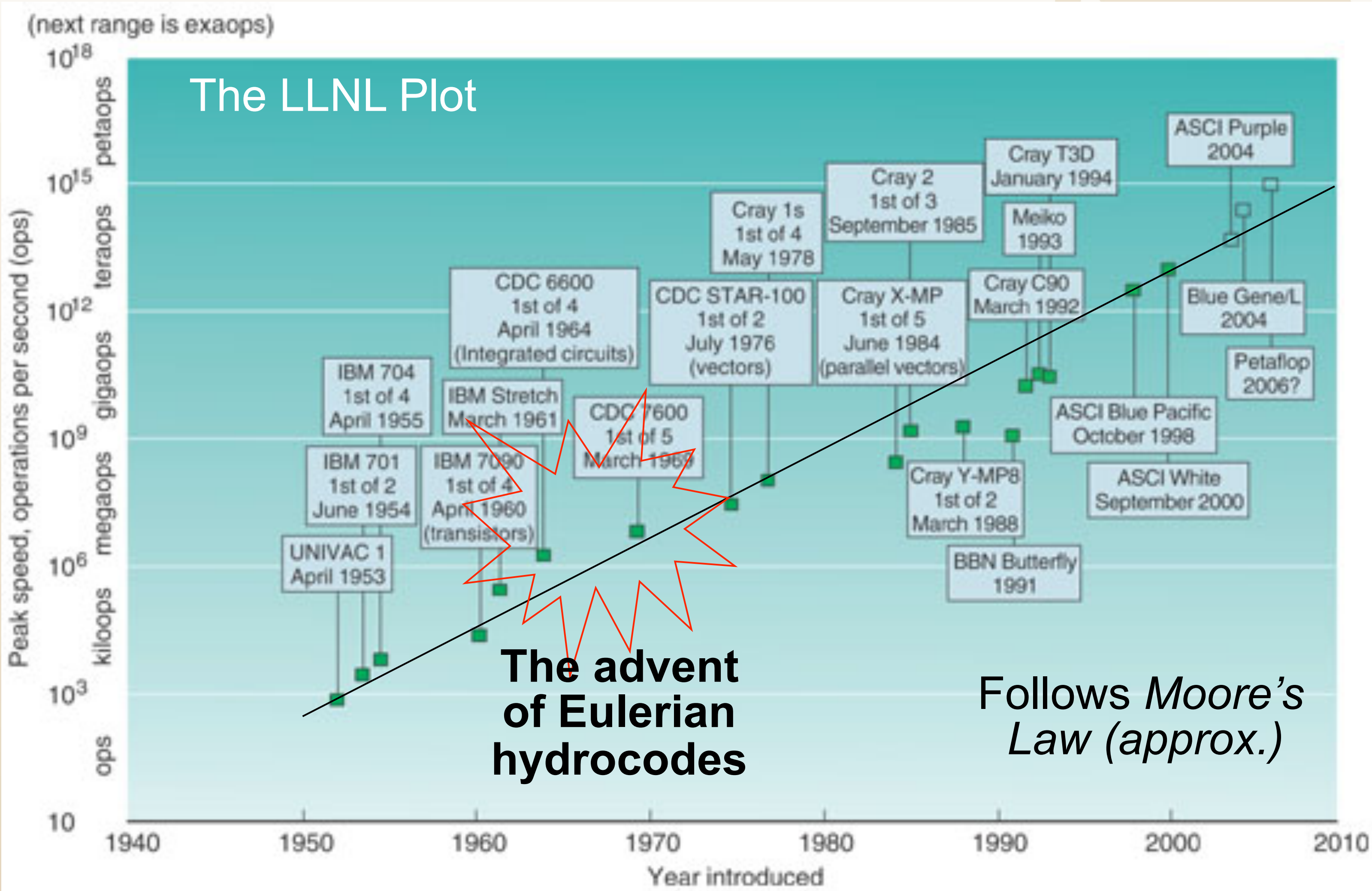
- How did you begin programming?
- The first programming methods are terrifying to behold!



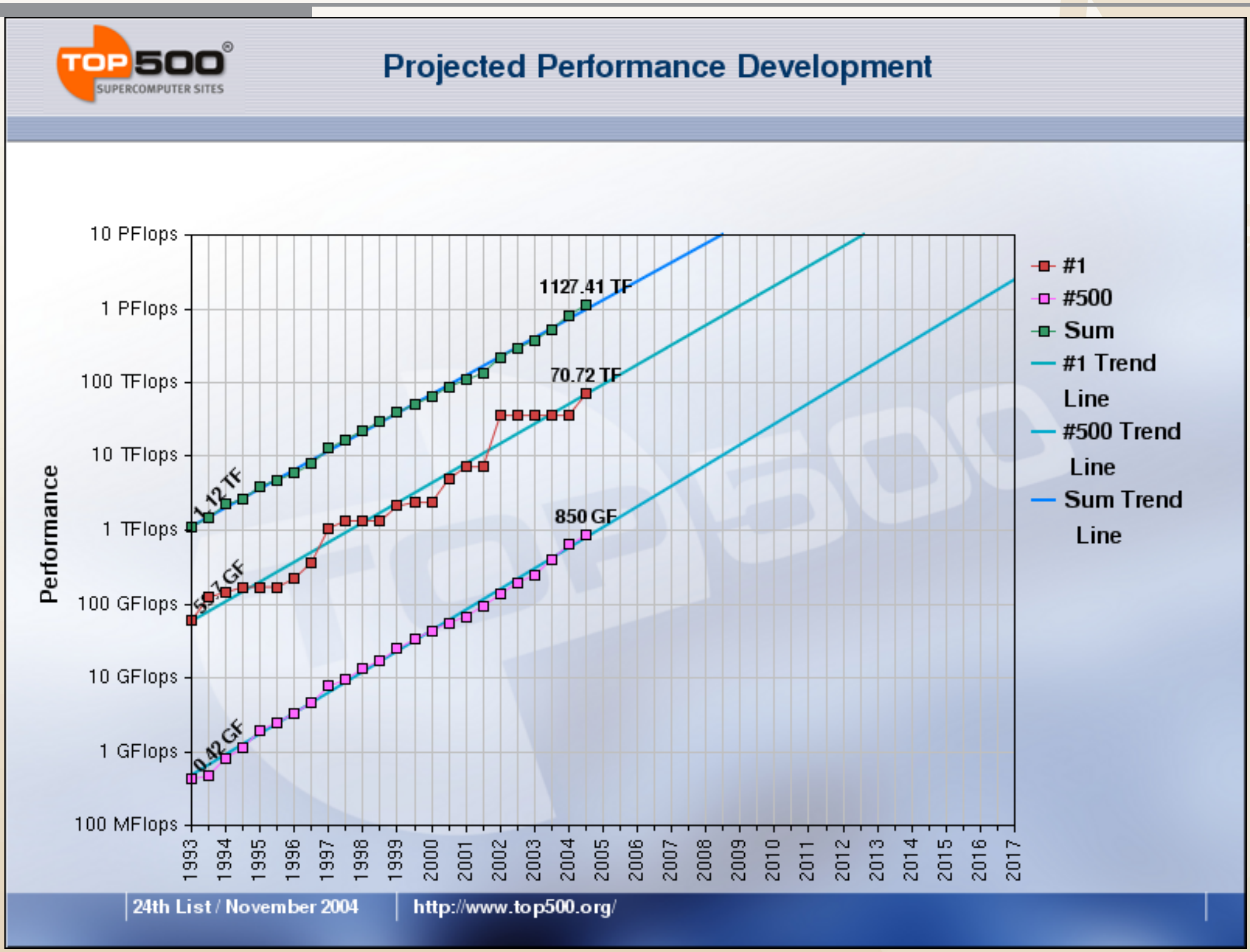
Programmer's Reference Manual

THE FORTRAN AUTOMATIC CODING SYSTEM FOR THE IBM 704 EDPM

The most obvious aspect is the raw performance of the machines.

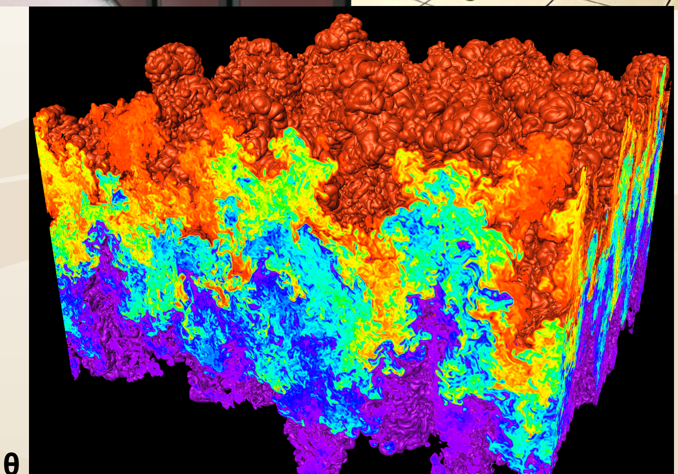


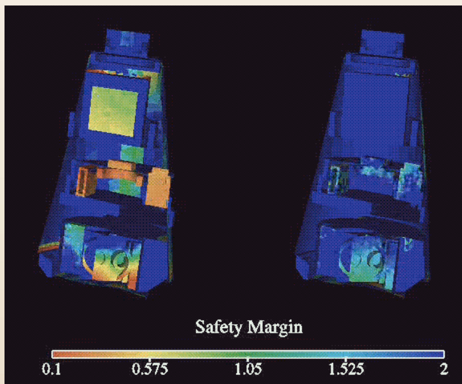
More recently the whole World has played in this field

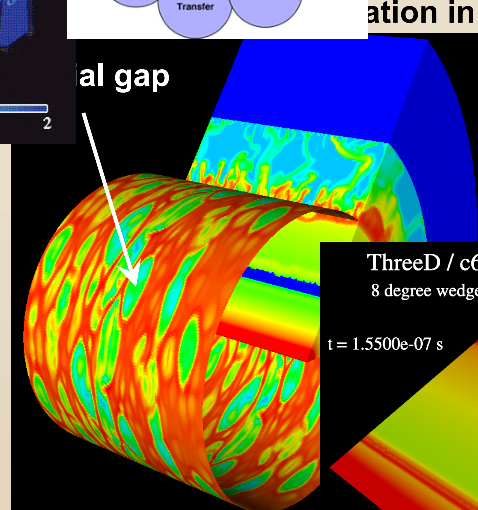


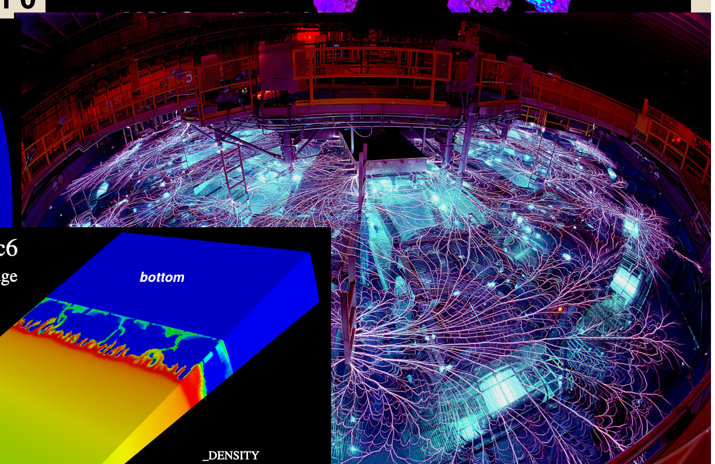
What about the future?

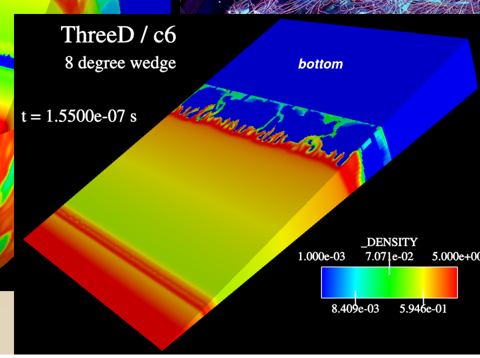
"But the only way of discovering the limits of the possible is to venture a little way past them into the impossible."
Arthur C. Clarke [Clarke's Second Law]











"Nothing is destroyed until it is replaced."
-Auguste Compte

NUMERICAL METHODS AND ANALYSIS FOR CFD

Quote by Peter Lax: The American Mathematical Monthly, February 1965:

"…*who may regard using finite differences as the last resort of a scoundrel that the theory of difference equations is a rather sophisticated affair, more sophisticated than the corresponding theory of partial differential equations*."

He goes on to make two points:

1. The proofs that an approximation converges is analogous to the estimates of the soln's to the PDEs (points to the CFL paper in 1928)*
2. These proofs are harder to construct than for the PDEs

***CFL=Courant, Friedrichs, Lewy which used numerics to prove the existence of soln's to PDE and gives us the term CFL condition.**

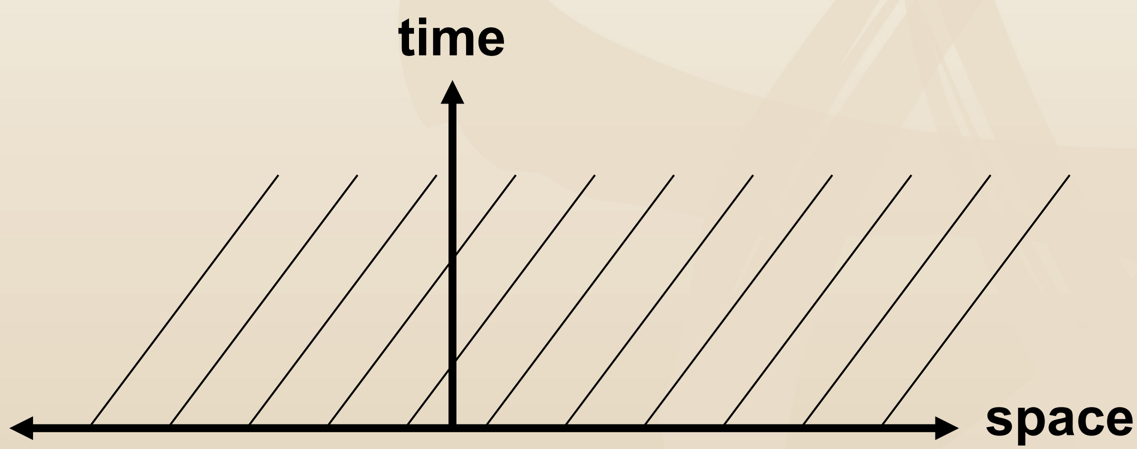
The scalar wave equation is both simple, but challenging to numerical schemes.

- Its as simple as a PDE can get $\frac{\partial \rho}{\partial t}+a\frac{\partial \rho}{\partial x}=0$, usually $a=1$

  hyperbolic
- Its solution is simple too $\rho\left(x,t\right)=\rho\left(x-at,0\right)$
- It introduces a number of important concepts and makes a great test for codes and methods.
- Its a lot harder to solve numerically than one might think.
- Jay Boris describes this equation as an “embarrassment to computational physics”
  - This does not apply to special methods.
  - This is because despite its complexity, methods that are general (apply to more complex equations and systems) have trouble solving the linear advection equation well.

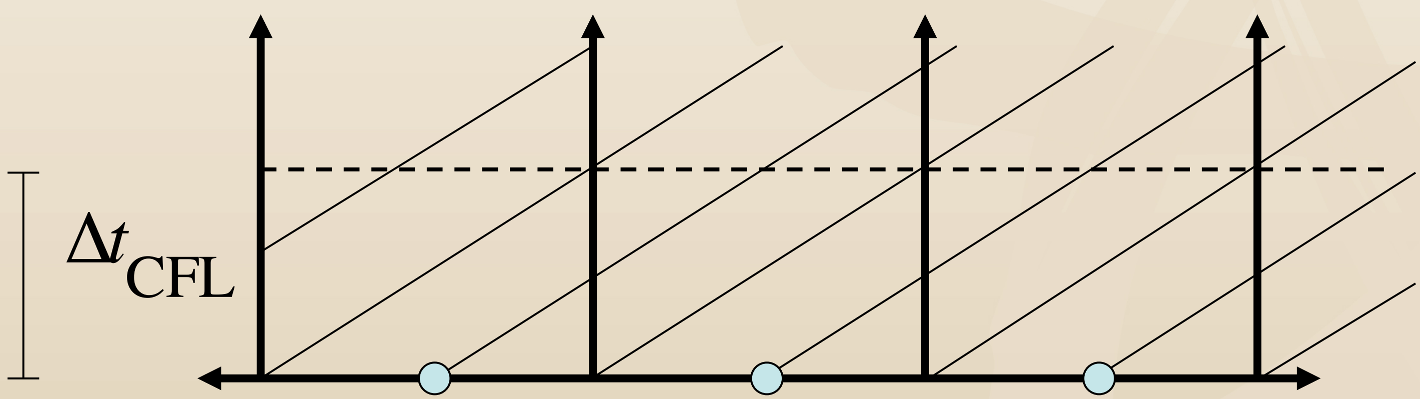
Characteristics are an incredibly useful concept.

- This equation provides a painless introduction to characteristics, the rays in space-time that solutions follow.
- With the linear wave equation the characteristics are straight.



Domain of dependence of a solution leads directly to the Courant or CFL number.

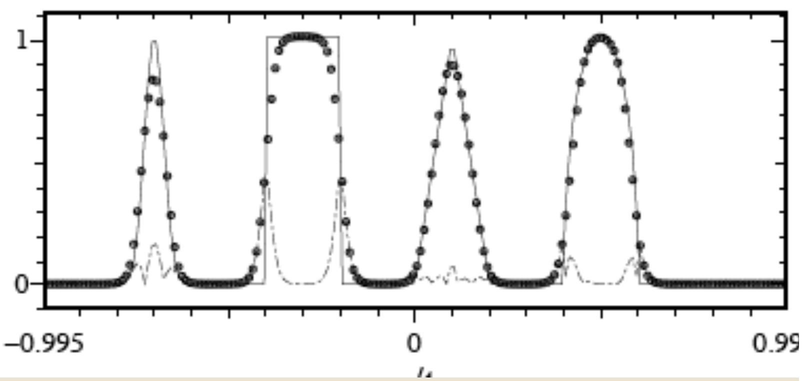
- This is the region of space that can be physically effected by another space due to the finite speed of propogation.
- The idea originated with Courant, Friedrichs and Lewy in 1928 related to the analytic existence of solutions to PDEs (discretization was used as a device in the proofs).

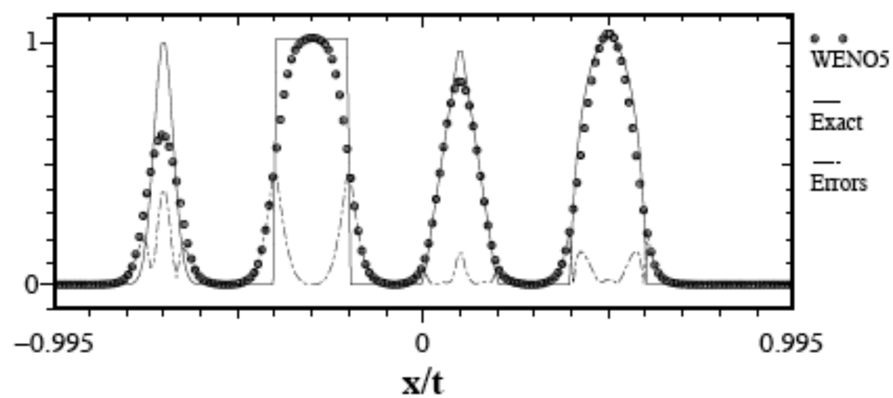


Smoothness is an important aspect of the profiles advected with this equation.

- Even though the profile is not modified by the exact differential equation, the numerical solution will disturb the profile according to two conditions,
  - The nature of the numerical method itself,
  - And the nature of the profile.
- Numerical methods have much more pathological behavior near **discontinuous** solutions
- Conversely, one can only expect ideal behavior for **smooth** solutions.
- The nature of the scalar advection equation means that *once an error is made, its kept in the numerical solution forever*.
  - This is at the heart of the difficulty of solving this equation numerically.

An example of the scalar wave equation in action: Rider, Greenough & Kamm 2005.





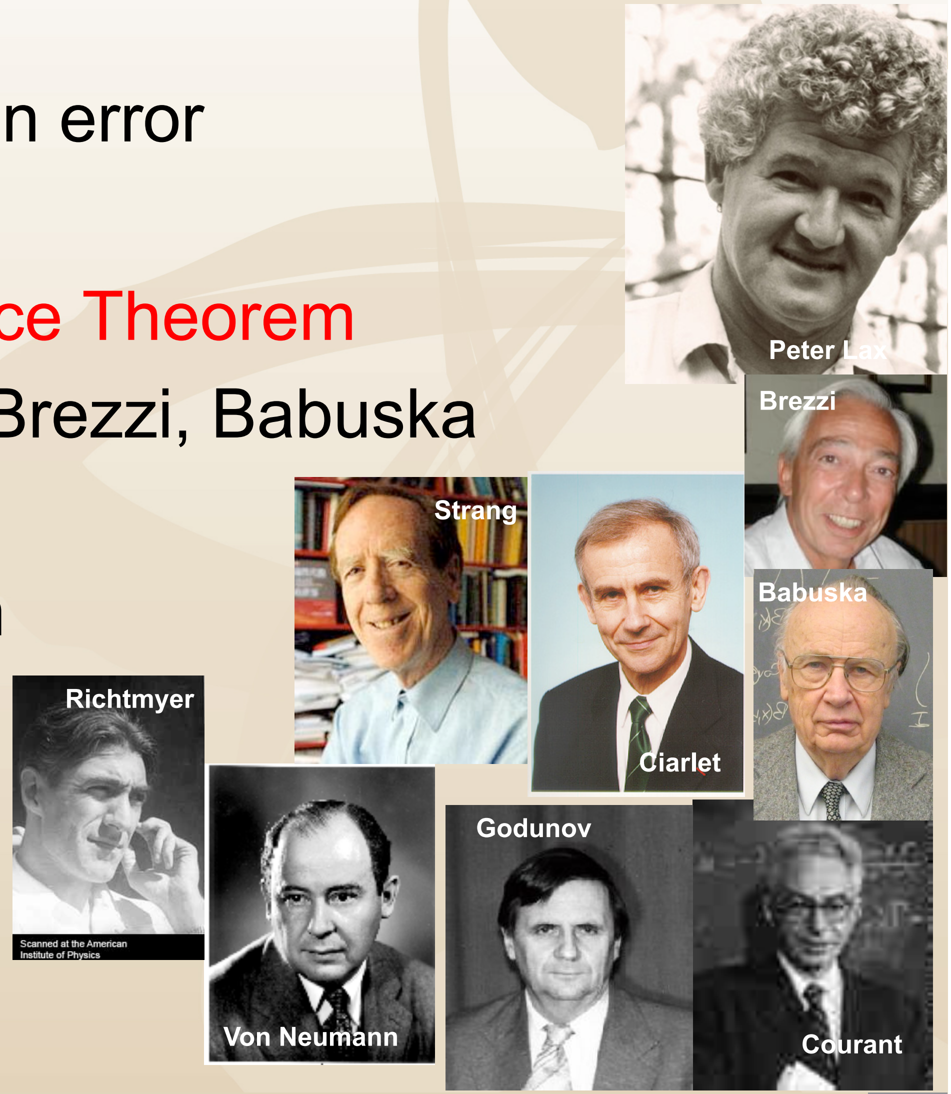
Method	Grid	L_1	L_1 rate	L_2	L_2 rate	L_∞	L_∞ rate
WENO5	16	2.64×10^{-3}		3.30×10^{-3}		6.84×10^{-3}	
	32	1.13×10^{-4}	4.55	1.51×10^{-4}	4.45	3.64×10^{-4}	4.23
	64	5.10×10^{-6}	4.47	6.51×10^{-6}	4.54	1.90×10^{-5}	4.26

The basic theoretical expectations are essential to understand...

- Truncation or approximation error
- Stability
- Lax (Richtmyer) Equivalence Theorem
- FEM: Strang&Fix, Ciarlet, Brezzi, Babuska

In hyperbolic PDEs

- The Lax-Wendroff theorem
- Godunov's theorem
- Entropy conditions
- The LeFloch-Hou theorem



Local truncation error is the most basic concept in numerical approximation

- This can be estimated with the aid of a Taylor series expansion.

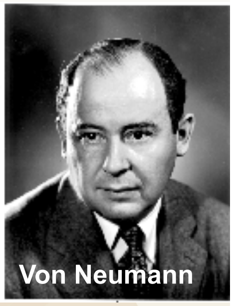
$$\exp(at) \underset{t\to 0}{\approx} 1 + at + \frac{a^2t^2}{2} + \frac{a^3t^3}{6} + \frac{a^4t^4}{24} + \cdots \frac{a^nt^n}{n!}$$

- This measures the difference between the discrete and continuous versions of the equations.

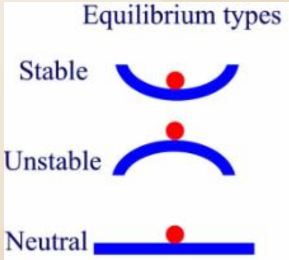
$$\textbf{truncation error} \underset{h\to 0}{=} \textbf{exact - numerical}$$

- When combined with stability it forms the foundation of numerical analysis.

Numerical stability is central to successful methods.



- A stable approximation is a pre-requisite for the use of that approximation.
- We introduce the basic concept with the analysis of a simple ODE integrator.
- An amplification factor is used to describe the stability of a method (greater than one is bad! Although less than one implies damping.)
- *Basically, one desires that the amplification of errors will be bounded, which usually means they will be damped!*



We can examine the basic stability concepts with ODEs.

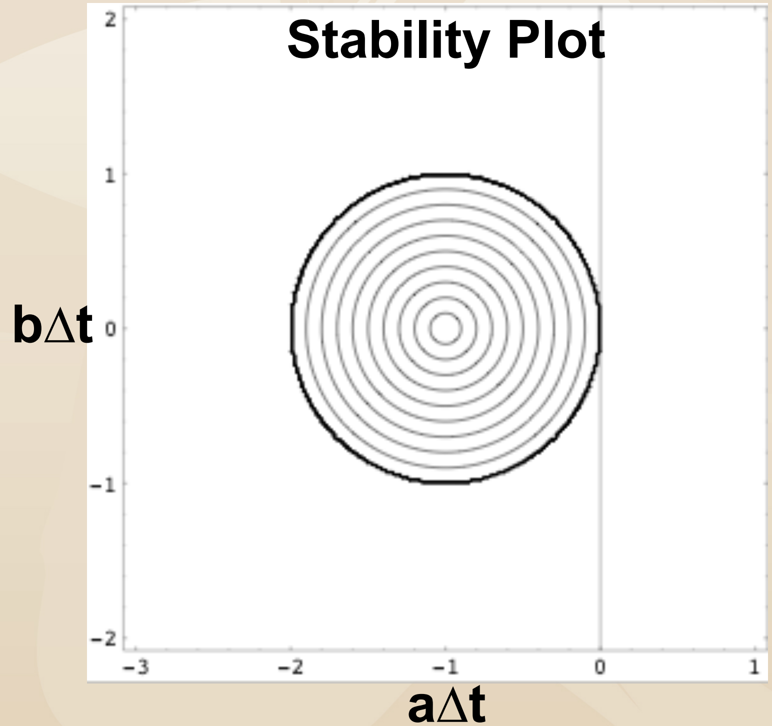
- The forward Euler example.

$$\frac{U^{n+1}-U^{n}}{\Delta t}=L\left(U^{n}\right)\rightarrow U^{n+1}=U^{n}+\Delta tL\left(U^{n}\right)$$

$$L=a+bi$$

- Truncation error

$$\frac{\Delta t^{2}}{2}\frac{\partial^{2}L\left(U\right)}{\partial t^{2}}+\frac{\Delta t^{3}}{6}\frac{\partial^{3}L\left(U\right)}{\partial t^{3}}+\text{H.O.T.}$$



The Lax-Richtmyer equivalence theorem provides the barest requirements on methods.

- Putting numerical stability and truncation error together gets us to the basic requirement for linear methods for differential equations.

Theorem (Lax Equivalence): ***A numerical method for a linear differential equation will converge if that method is consistent and stable. Comm. Pure. Appl. Math. 1954***

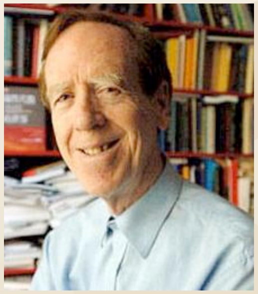
Consistency - means that the method is at least 1st order accurate – means it approximates the correct PDE.

Stable - the method produces bounded approximations

Important to recognize for its relation to verification.

Let’s state this differently (Gil Strang, Introduction to Applied Mathematics)

- The fundamental theorem of numerical analysis, The combination of consistency and stability is equivalent to convergence.



- There is a similar theorem for ODEs courtesy of Dahlquist, which applies to nonlinear functions!

Mathematical expectations for the numerical solution of elliptic and parabolic PDEs

- It is generally possible to get the design order of accuracy intended for these classes of PDEs due to smoothness.
- For general cases with discontinuities and singularities, it is still possible to get the full order accuracy, but…
  - The ability of a method to achieve this is dependent on the method's utilization of special features to deal with the difficulties.
  - Does the testing of the method provide confidence that the special features indeed provide this?

Lax-Wendroff Theorem is an essential motivator for many numerical methods.

- Most methods for hyperbolic PDEs are based on the discrete conservation form following the continuous conservation form because of this theorem.

Theorem (Lax and Wendroff): ***If a numerical method is in discrete conservation form, if a solution converges, it will converge to a weak solution of the PDE. A weak solution is not the weak solution. There are infinitely many weak solutions.***

Conservation form: the flux out of one cell is into another

$$\frac{\partial u}{\partial t}+\frac{\partial f\left(u\right)}{\partial x}=0\Rightarrow u_j^{n+1}=u_j^n-\frac{\Delta t}{\Delta x}\left(f_{j+1/2}-f_{j-1/2}\right)$$

The key to the Lax-Wendroff theorem is using conservative fluxes.

- This means putting the equations being solved in flux form,

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(f_{j+1/2} - f_{j-1/2}\right)$$

- In this form all the flux contributions will telescope or collapse into boundary terms.

$$\sum_{k=k_0,k_N} u_k^{n+1} = \sum_{k=k_0,k_N} u_k^n - \frac{\Delta t}{\Delta x}\left(f_{k+1/2} - f_{k-1/2}\right)$$
$$\Rightarrow \sum_{k=k_0,k_N}\left(u_k^n\right) - \frac{\Delta t}{\Delta x}\left(f_N - f_0\right)$$

- This mimics the behavior of physically conserved fluxes.

Here is an example of what happens without conservation form. Burgers' equation.

Nonconservation form

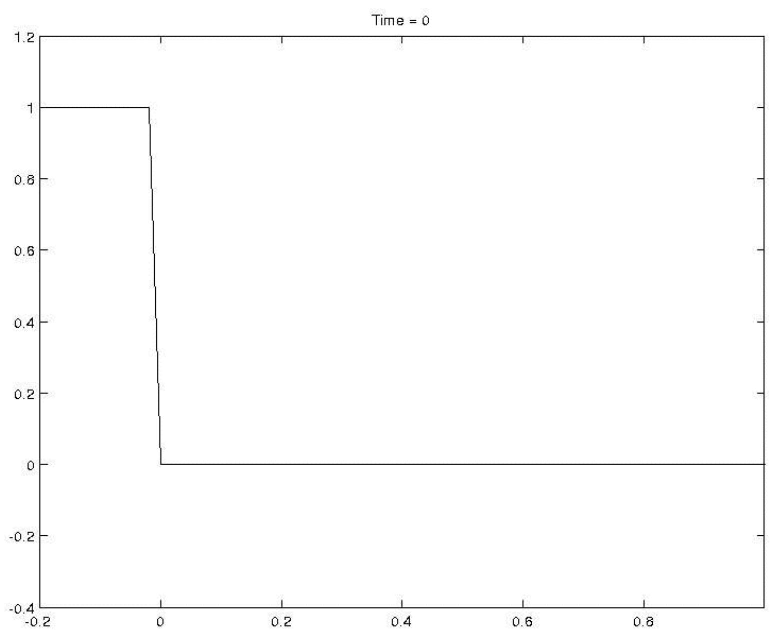
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

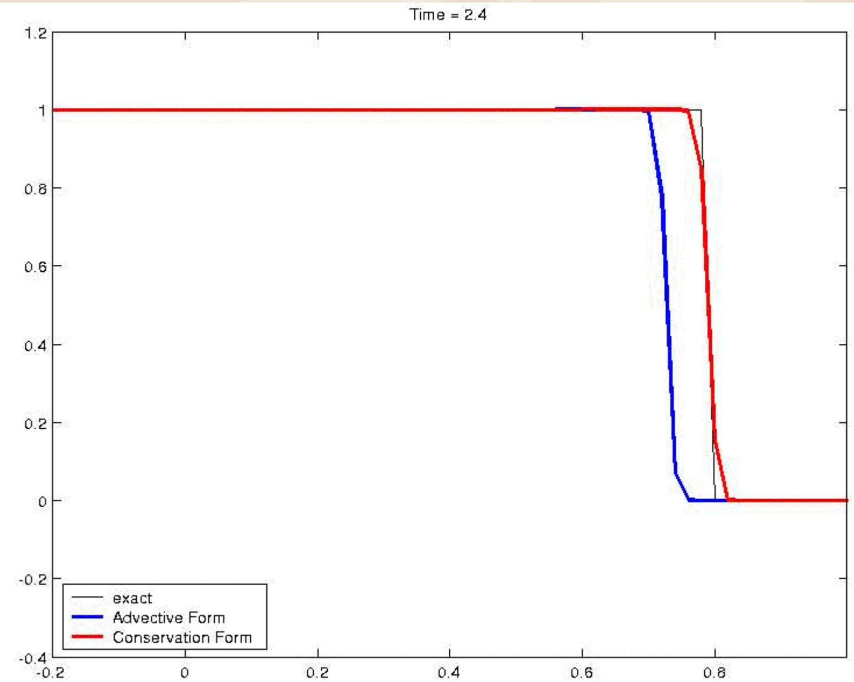
$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} u_j^n \left(u_j^n - u_{j-1}^n \right)$$

Conservation form

$$\frac{\partial u}{\partial t} + \frac{\partial\left(\frac{1}{2}u^2\right)}{\partial x} = 0$$

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(\tfrac{1}{2}\left(u_j^n\right)^2 - \tfrac{1}{2}\left(u_{j-1}^n\right)^2 \right)$$





Example from Randy Leveque

Entropy conditions are critical in determining physically meaningful results.

- The problem with L-W is that there are an infinity of weak solutions, we need a mechanism to pick out the correct physical one.
- The mechanism to do this entropy. The entropy created through dissipation, numerical viscosity.
- This is the connection to vanishing viscosity, more generally,

$$\frac{\partial u}{\partial t}+\frac{\partial f\left(u\right)}{\partial x}=\underset{\lambda\rightarrow 0^{+}}{\lambda\frac{\partial^2 u}{\partial x^2}}$$

Godunov’s theorem is critical to the development of modern methods.

- As mentioned earlier, it is a “barrier theorem” stating what cannot be done.
- It states that a linear second-order method cannot be monotone (i.e. non-oscillatory).
- The key word is “linear”.
- Modern methods are nonlinear and monotonicity-preserving. The nonlinearity makes the difference stencil dependent on the solution.

The Hou-LeFloch theorem has potentially profound consequences .

- What happens when the method is not in conservation form?
- The solution does not converge to a weak solution much less a correct one regardless of the dissipation.

<u>Theorem (Hou-LeFloch):</u> For a non-conservative method the solution differs from a weak solution by an amount proportional to the entropy produced in the solution. Math. Comp. 62, 1994

The Majda-Osher theorem establishes accuracy expectations for discontinuous flows.

- Majda and Osher establish that the approximation of shocked or discontinuous flows will converge at be 1st order at best.

Theorem (Majda and Osher): A numerical solution will converge at 1st order at best for the region between any characteristics emanating from a discontinuity. Comm. Pure Appl. Math. 1977

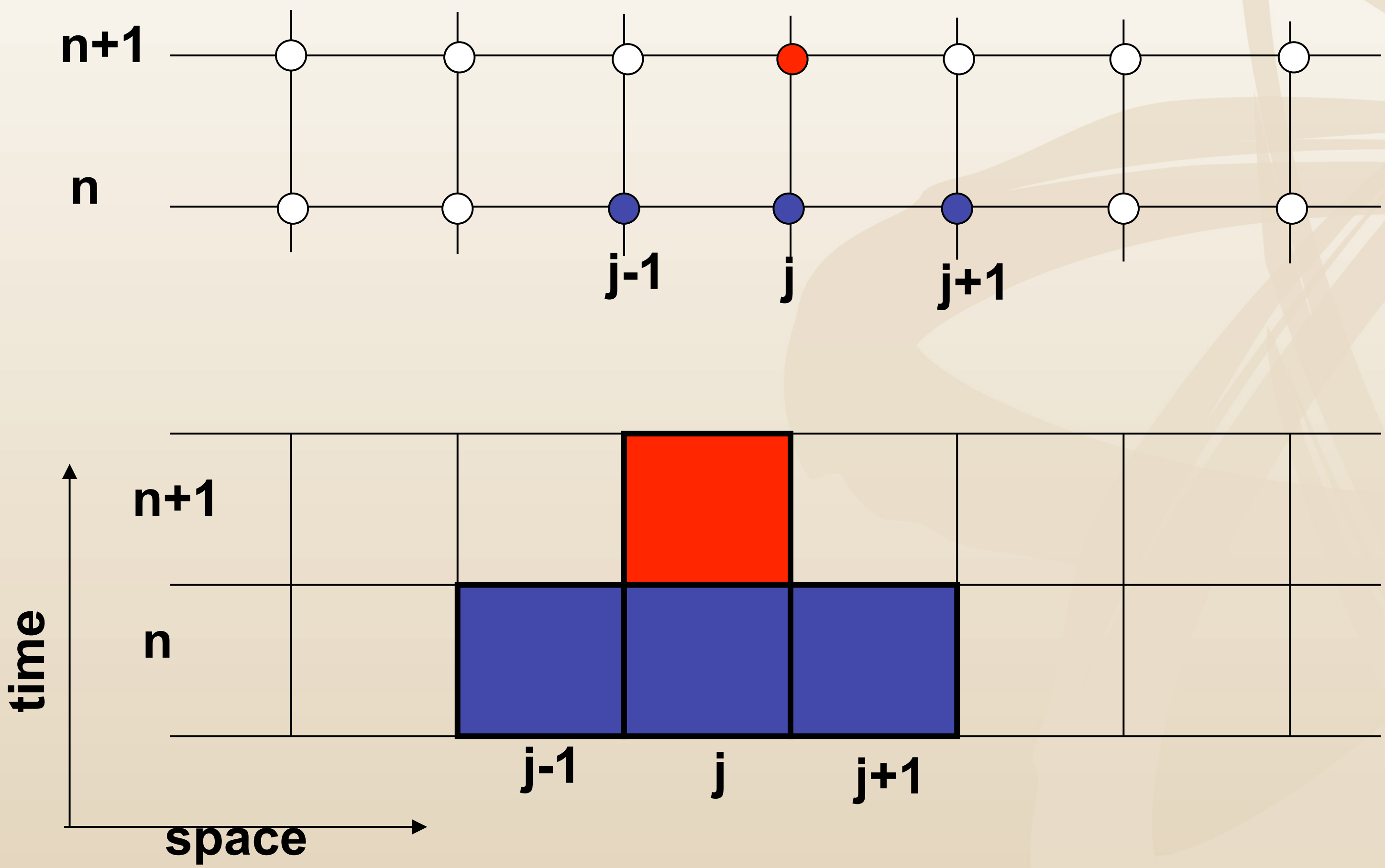
- Nonlinear discontinuties (self-steepening like shocks) converge at 1^{st} order.
- Linear discontinuties converge at less than 1^{st} order (order *m/(m+1)* where *m* is the order of the method, Banks, Aslam, Rider (2009))

Simple (linear) numerical methods

- Simple second-order differencing
- Artifical viscosity
- Upwind differencing
- Lax-Friedrichs
- Lax-Wendroff
- 2nd order upwinding
- Fromm's method

The methods here form the foundation for what follows. Advanced methods are put together from simplier ones.

Stencils are another way to express how methods are put together.



Let's start with the most obvious thing to try.

- We'll take the scalar wave equation and use centered differencing plus an forward Euler time stepping,

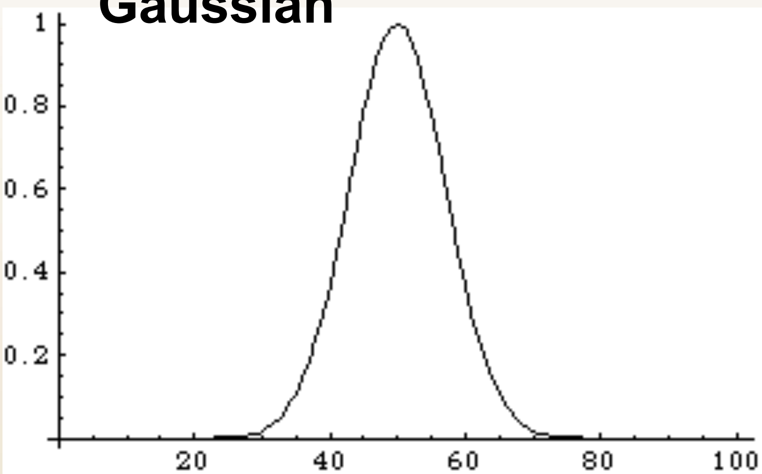
$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=0 \Rightarrow \frac{u_j^{n+1}-u_j^n}{\Delta t}+\frac{u_{j+1}^n-u_{j-1}^n}{2\,\Delta x}$$

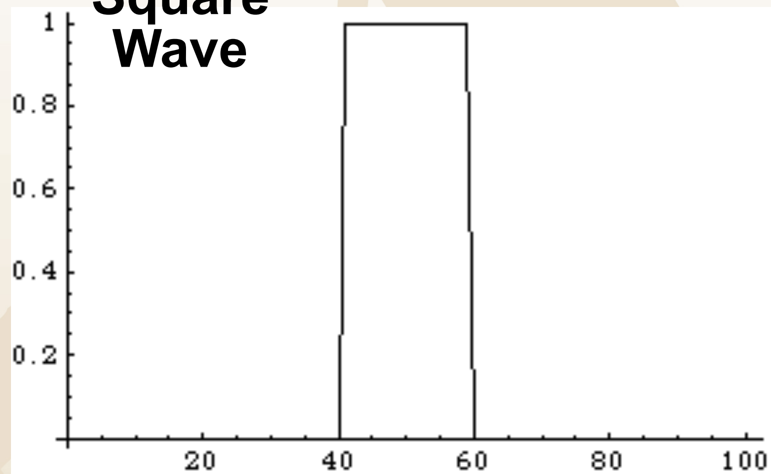
- The problem is that this method is unconditionally unstable.
- How can this be fixed?

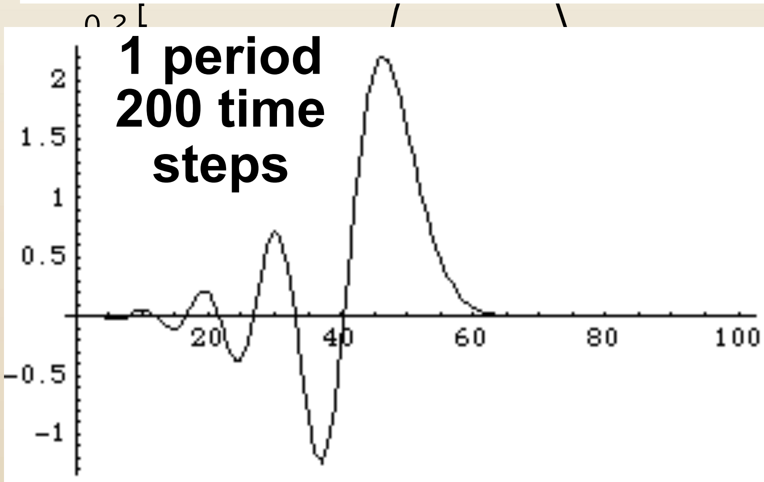
$$u_j^{n+1}=u_j^n-\Delta t\left(\frac{u_{j+1}^n-u_{j-1}^n}{2\,\Delta x}\right)$$

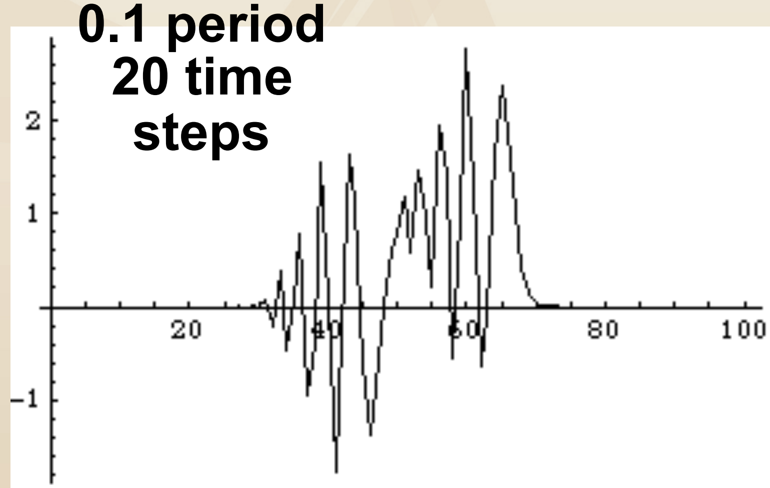
What does the instability look like?

I.C









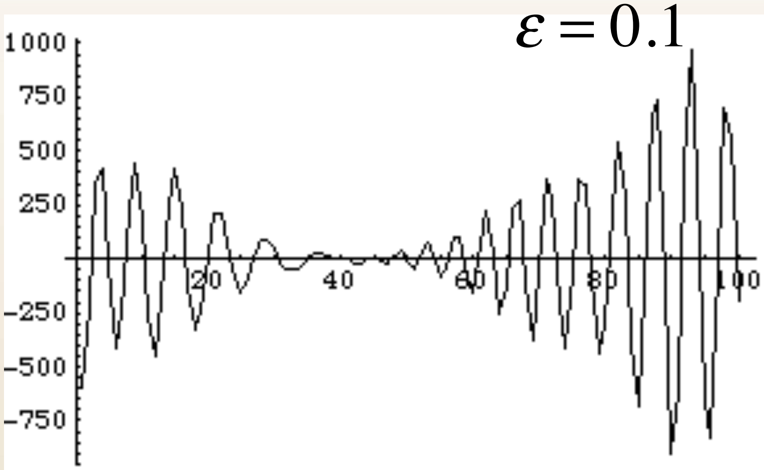
Artificial viscosity has its origin at Los Alamos

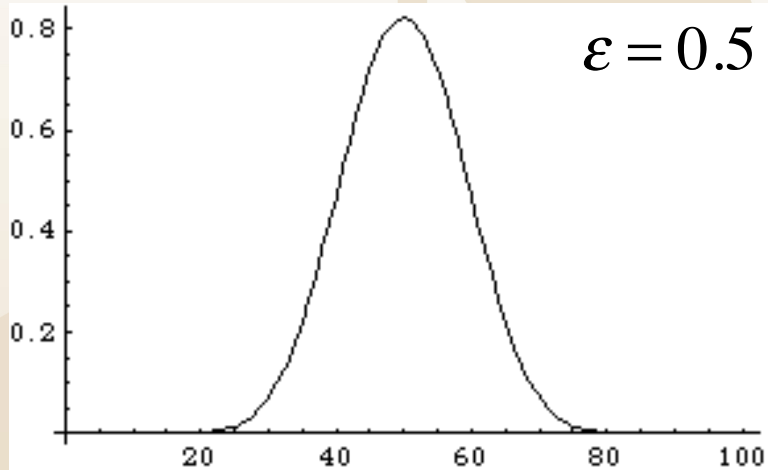
- The first open paper is by Von Neumann and Richtmy (1950, J. Appl. Phys.)- Richtmyer published a LA repo years earlier.
- I gave that report to you during Lecture 1.
- How about trying artificial viscosity to fix the simple differencing scheme!
- Let’s do an experiment and try different sizes of viscosity.

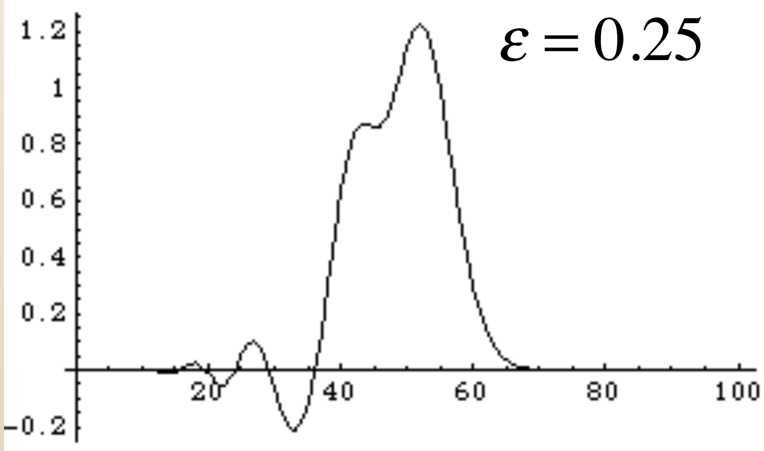
$$\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}=\varepsilon\frac{\partial^2 u}{\partial x^2}\Rightarrow\ u_j^{n+1}=u_j^n-\Delta t\left(\frac{u_{j+1}^n-u_{j-1}^n}{2\,\Delta x}\right)+\varepsilon\,\Delta t\left(\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{\Delta x^2}\right)$$

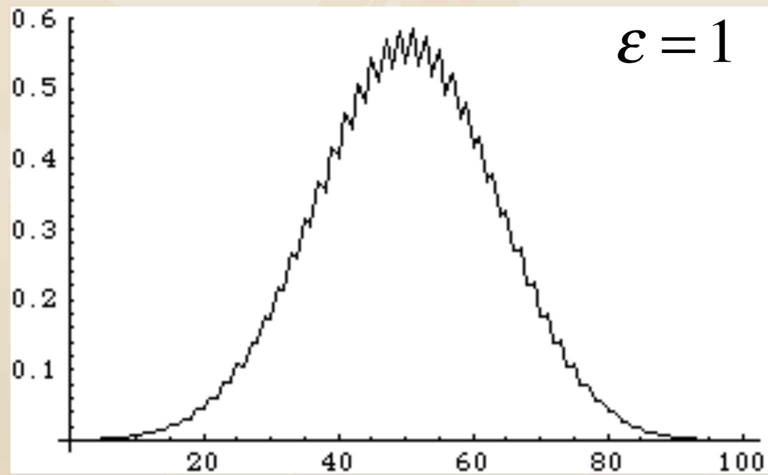
Results using artificial viscosity to stabilize the simple method.

$$\nu = \frac{\varepsilon \, \Delta t}{2 \, \Delta x^2}$$



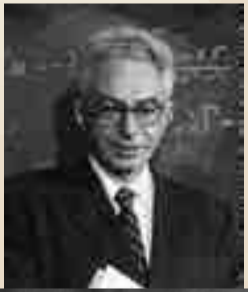






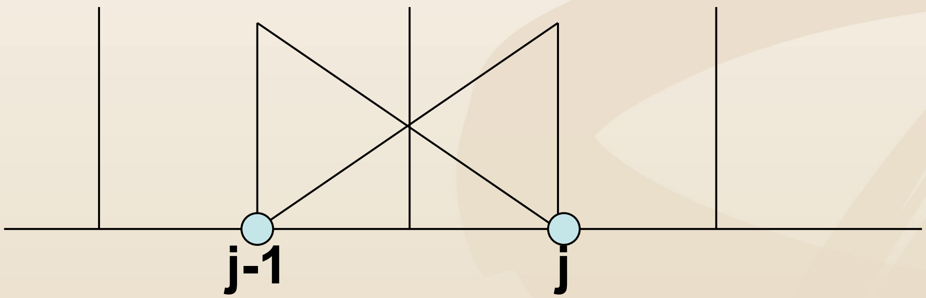
Upwind differencing is the archetypal CFD method (in the Eulerian-frame).

- That was not very satisfying, we can add some physics to the problem and stabilize the methods in a more satisfactory manner.
- If we consider the direction of wave propagation in constructing the differencing, upwind differencing results.
- This was originally introduced in 1952 by Courant, Issacson and Rees.
- Godunov also introduced another upwind form.



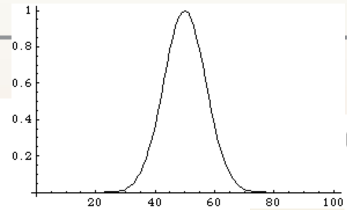
There are several ways to derive upwind differencing, here is the interpolation method.

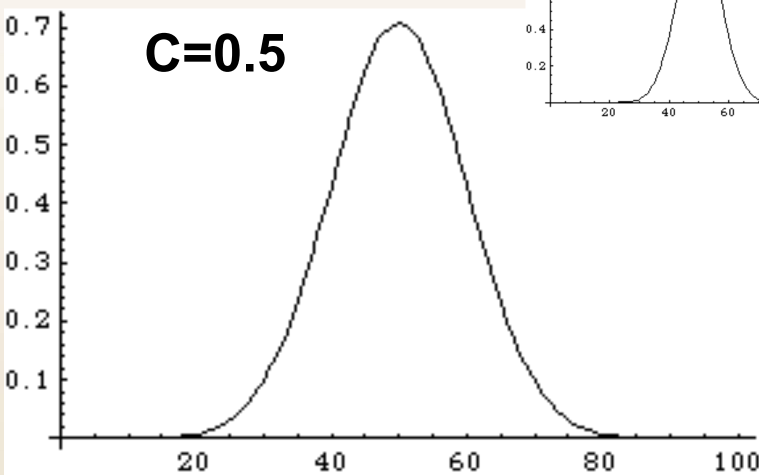
- The simplest way to get upwind differencing is interpolate back along characteristics using the CFL number

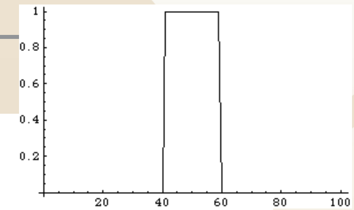


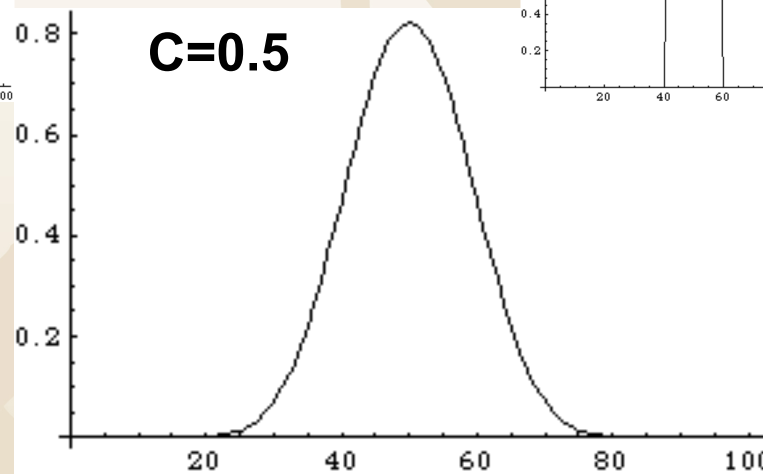
$$u_j^{n+1} = \left(1 - c\right) u_j^n + c u_{j-1}^n ; c = a \frac{\Delta t}{\Delta x}$$

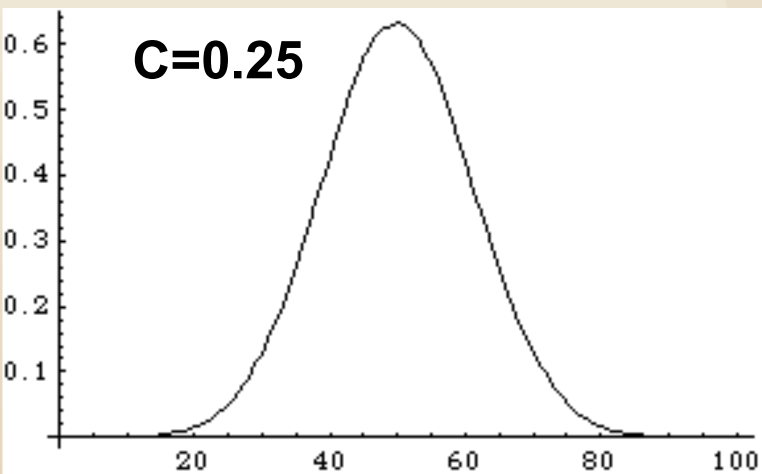
Let’s experiment with this formulation as before.

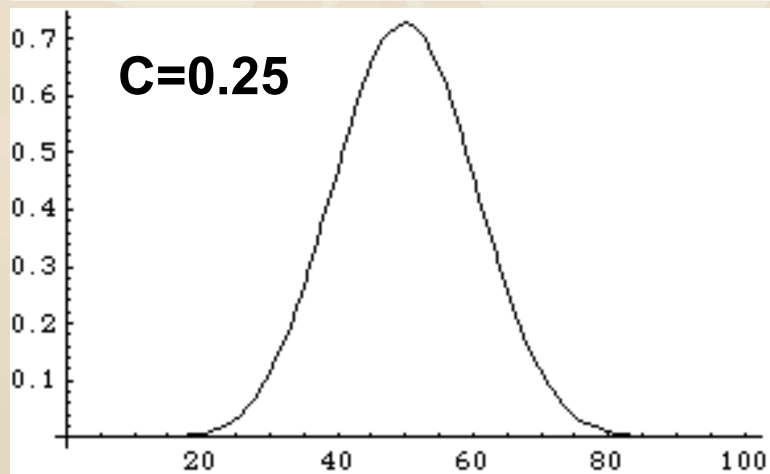




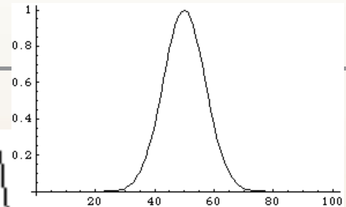


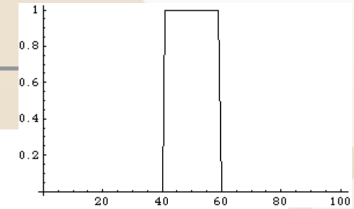


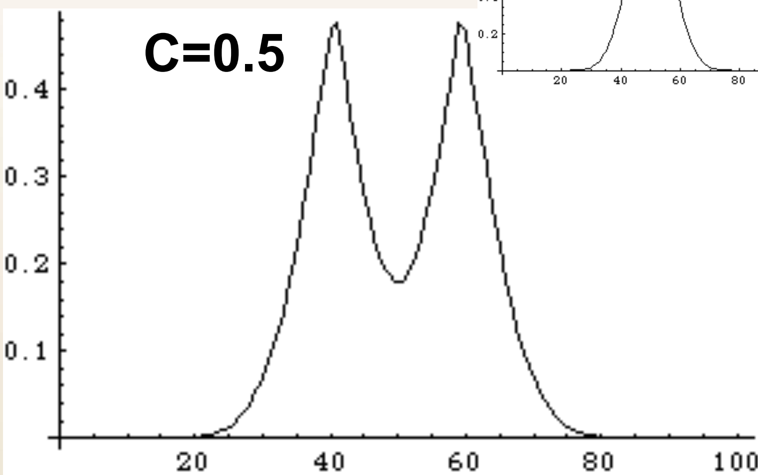


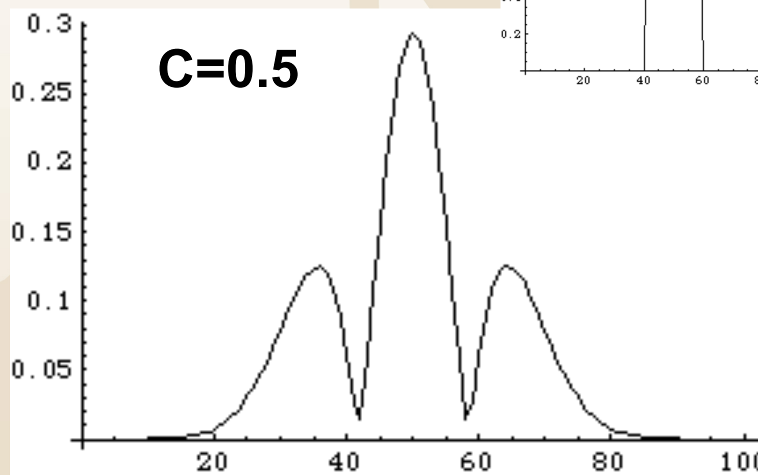


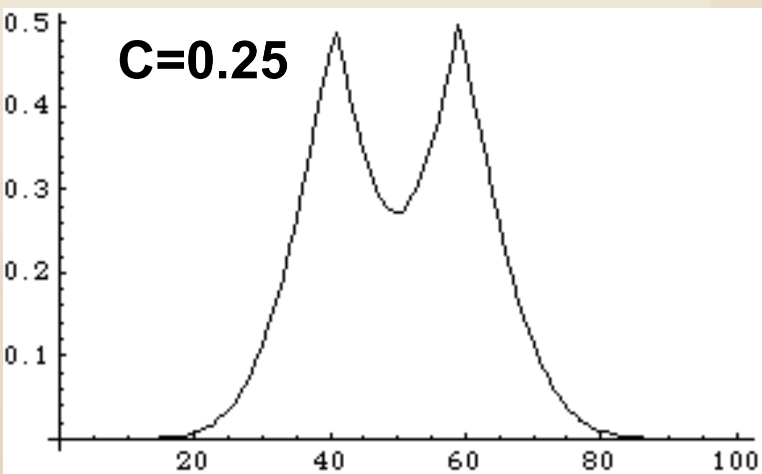
Examine the error's associated with the solutions.

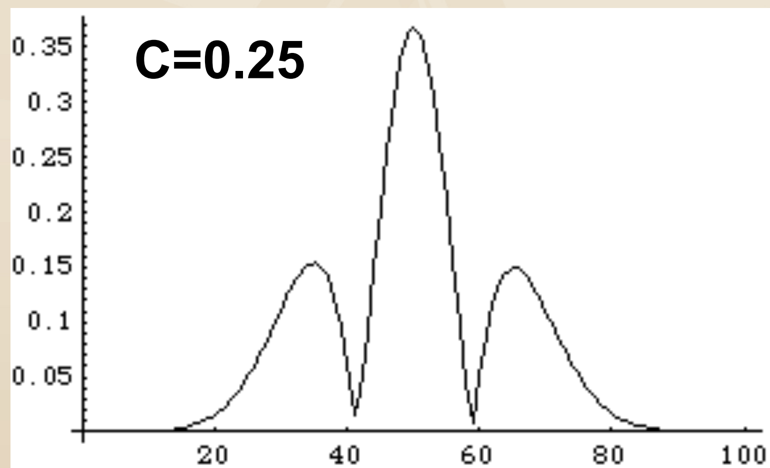












A couple of additional comments…

- This turns out to be the same as the artificial viscosity coefficient set to 0.5
- If this method is run with the CFL number equal to one, the exact solution is recovered.
  - The characteristic condition
- This method is first-order accurate in time and space. It is stable to a CFL number of one (proven in the next lecture).

Lax-Wendroff's method was a major development in computations.

- As well as be important theoretically, the joint paper produced a landmark method.
- The method is second-order accurate, stable to a CFL number of one.
- It corresponds to our artificial viscosity example of a coefficient of 0.25.
- The method is derived by expanding the solution in a Taylor series and substituting second-order approximations.

The derivation of the Lax-Wendroff method.

- Expand the function in a Taylor series,

$$u_j^{n+1} = u_j^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} + \text{H.O.T.}$$

- Assume the functions are smooth enough to exchange space and time derivatives,

$$u_j^{n+1} = u_j^n - \Delta t \frac{\partial u}{\partial x} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial x^2} + \text{H.O.T.}$$

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\,\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{\Delta t^2}{2\,\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

Lax-Wendroff method can be done as a two-step method.

- This was formulated by Richtmyer in 1963.

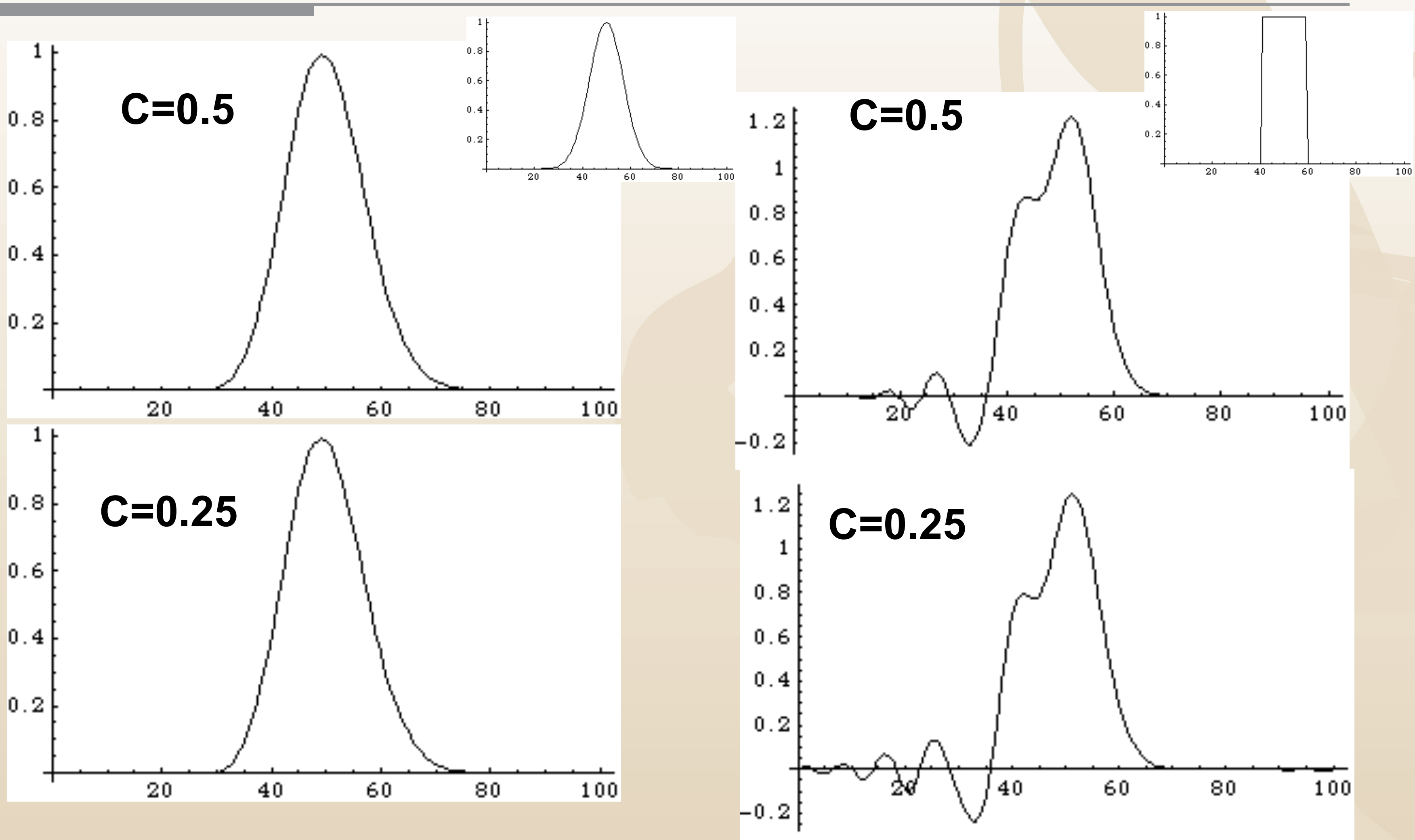
$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\,\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{\Delta t^2}{2\,\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

$$\Downarrow$$

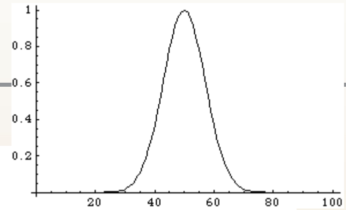
$$u_j^{n+1} = u_j^n - c\left(u_{j+1/2}^{n+1/2} - u_{j-1/2}^{n+1/2}\right)$$

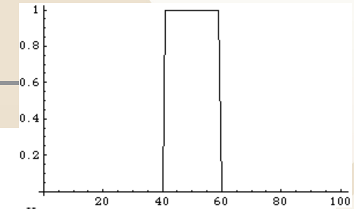
$$u_{j+1/2}^{n+1/2} = \frac{1}{2}\left(u_j^n + u_{j+1}^n\right) - \frac{\Delta t}{2\,\Delta x}\left(u_{j+1}^n - u_j^n\right)$$

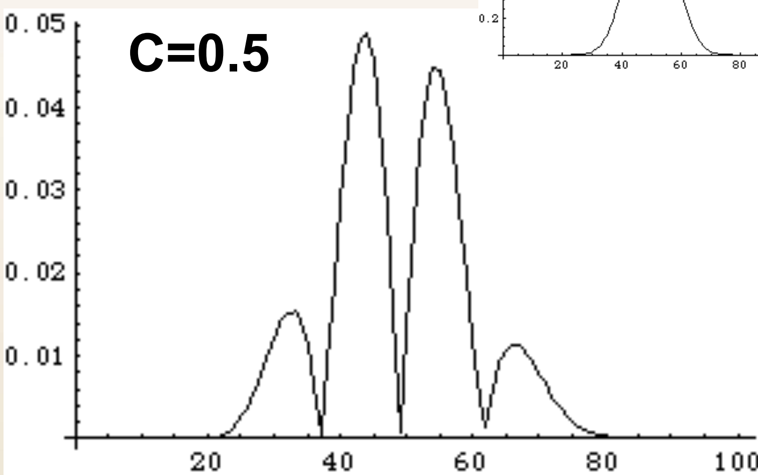
Let’s test the method as before…

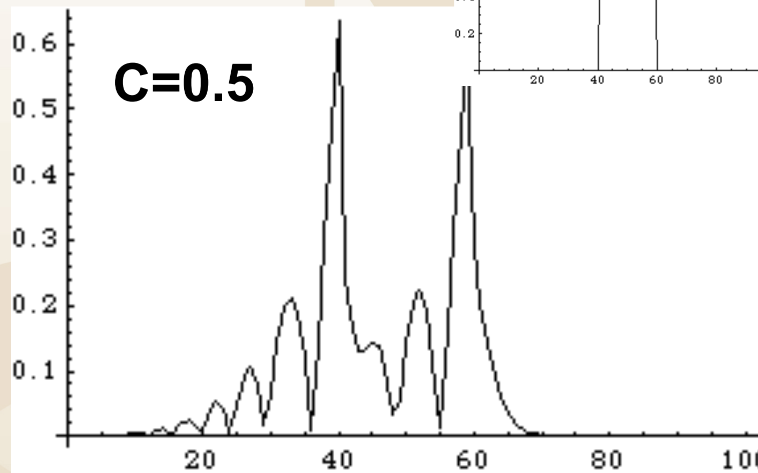


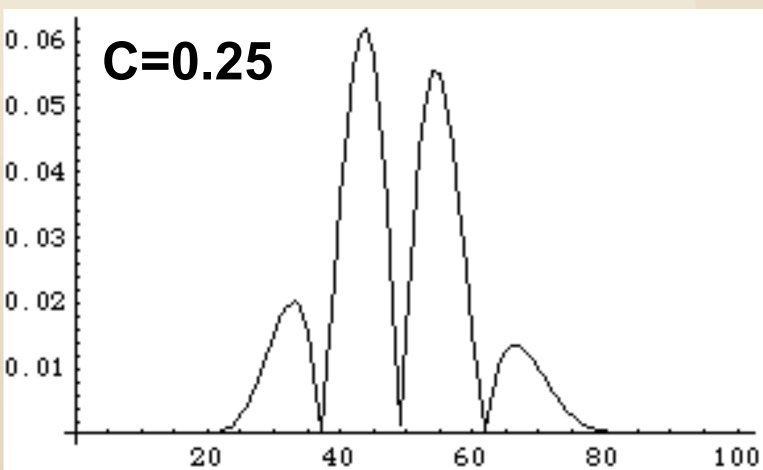
Examine the error’s associated with the solutions.

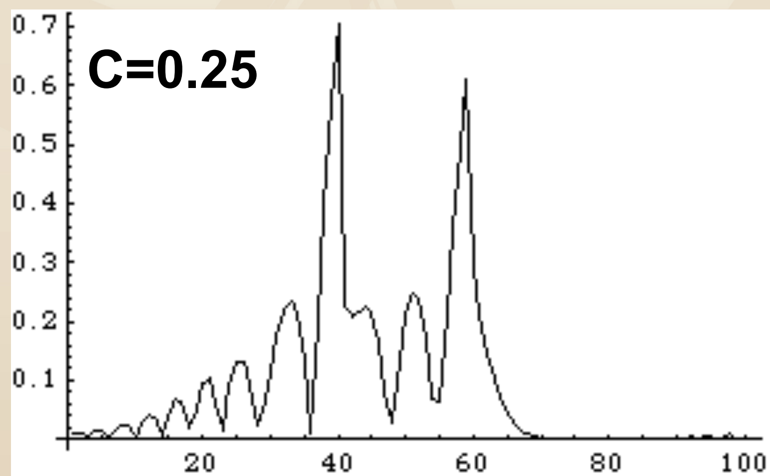












Second-order upwinding is the extension to upwinding but using a quadratic interpolation.

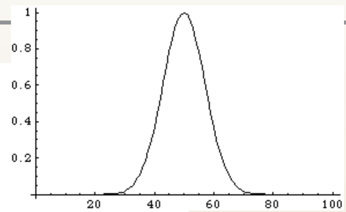
$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\,\Delta x}\left(u_j^n - 4u_{j-1}^n + 3u_{j-2}^n\right) + \frac{\Delta t^2}{2\,\Delta x^2}\left(u_j^n - 2u_{j-1}^n + u_{j-2}^n\right)$$

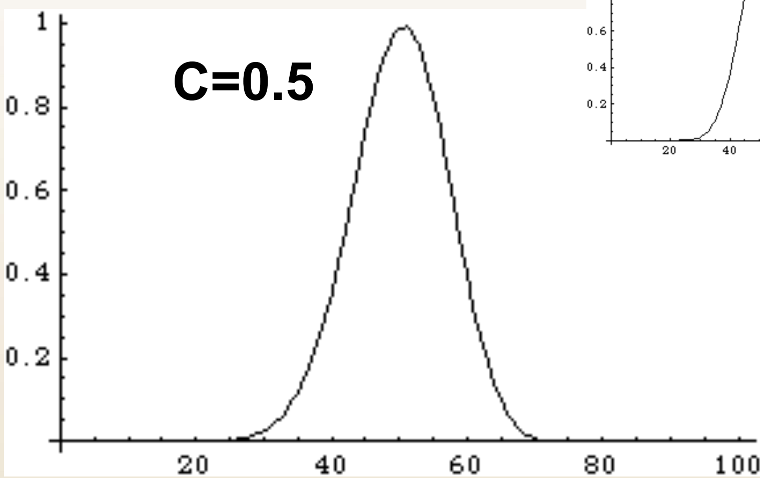
$$\Downarrow$$

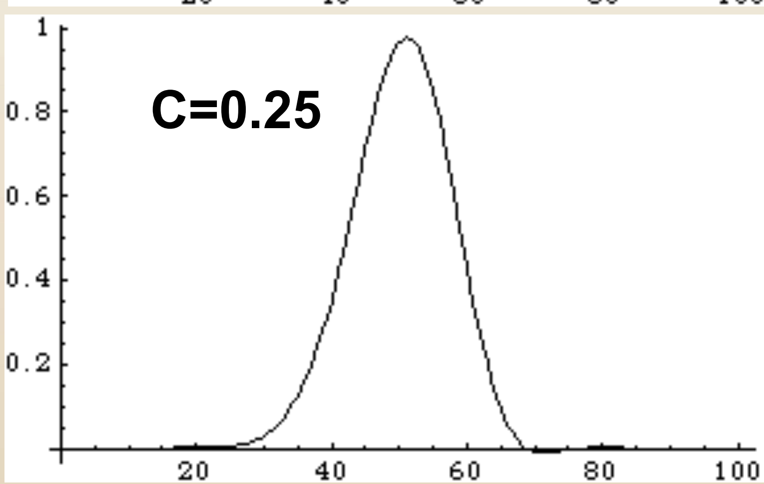
$$u_j^{n+1} = u_j^n - c\left(u_{j+1/2}^{n+1/2} - u_{j-1/2}^{n+1/2}\right)$$

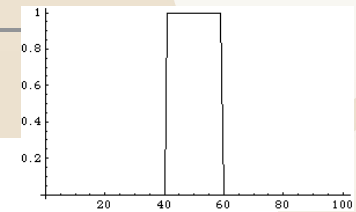
- It can also be derived using a Godunov-type of form by assuming each zone has a linear variation associated with it. The linear variation is upwind biased. (derived as part of the homework!)

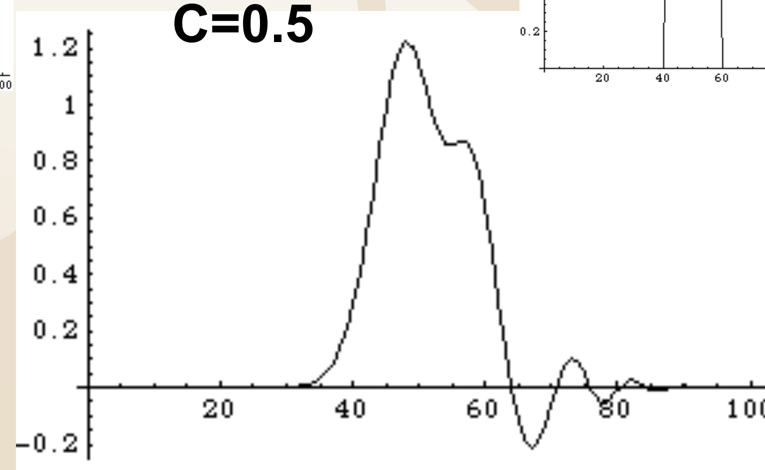
Let’s do the tests as before.

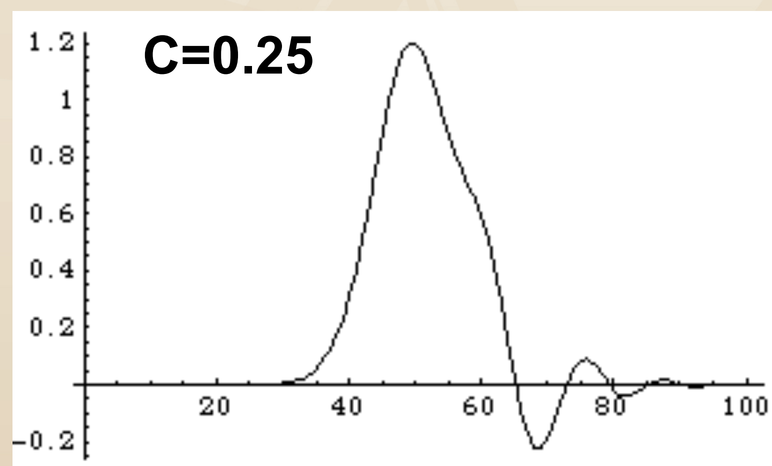




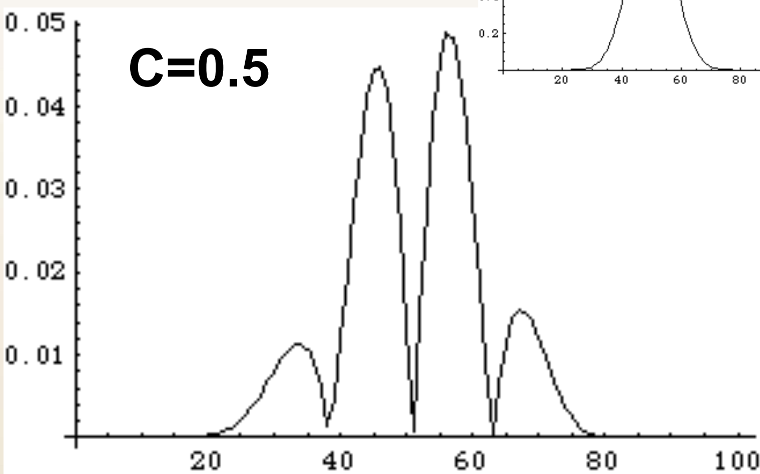


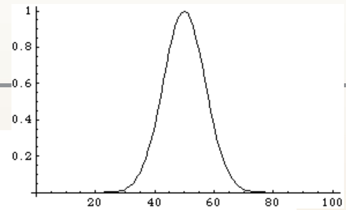


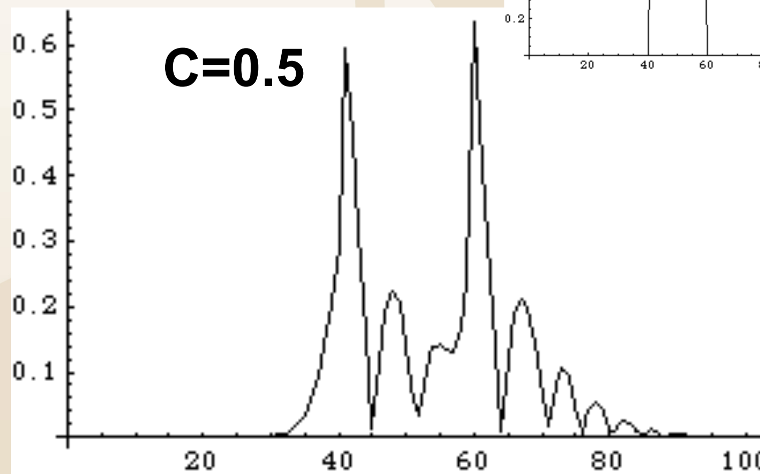


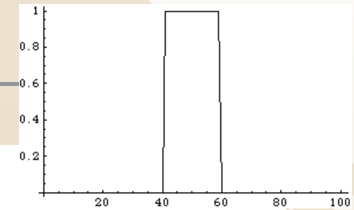


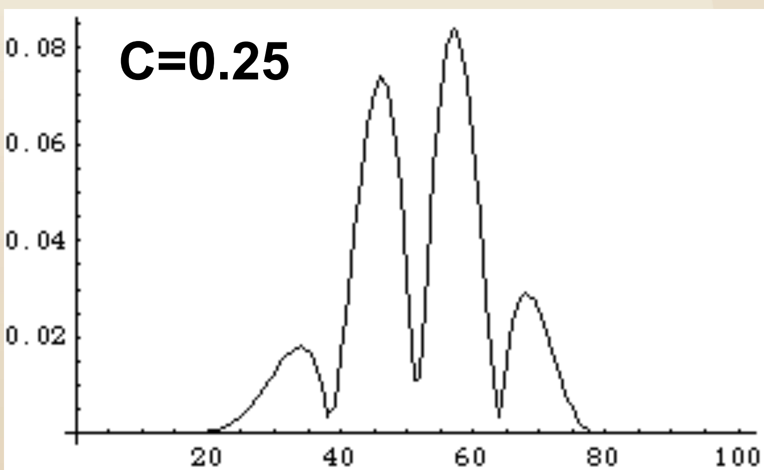
Examine the error's associated with the solutions.

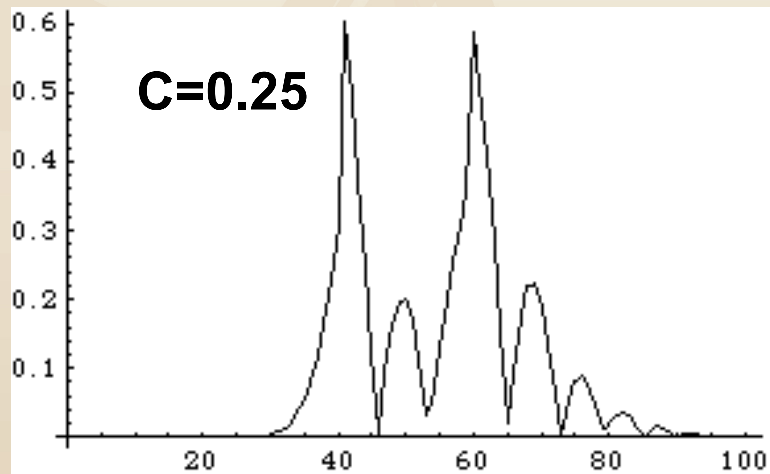












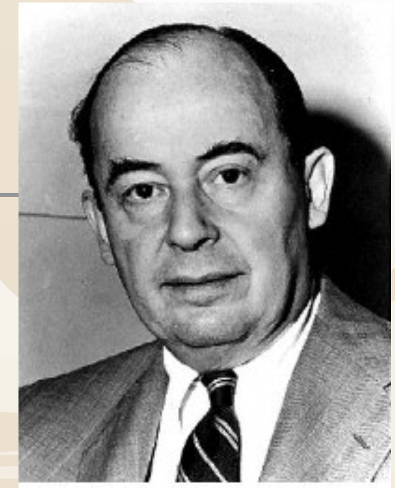
The stability, consistency and accuracy of the method can be studied via Fourier analysis.

- It involves substituting an analytical Fourier series for the grid function

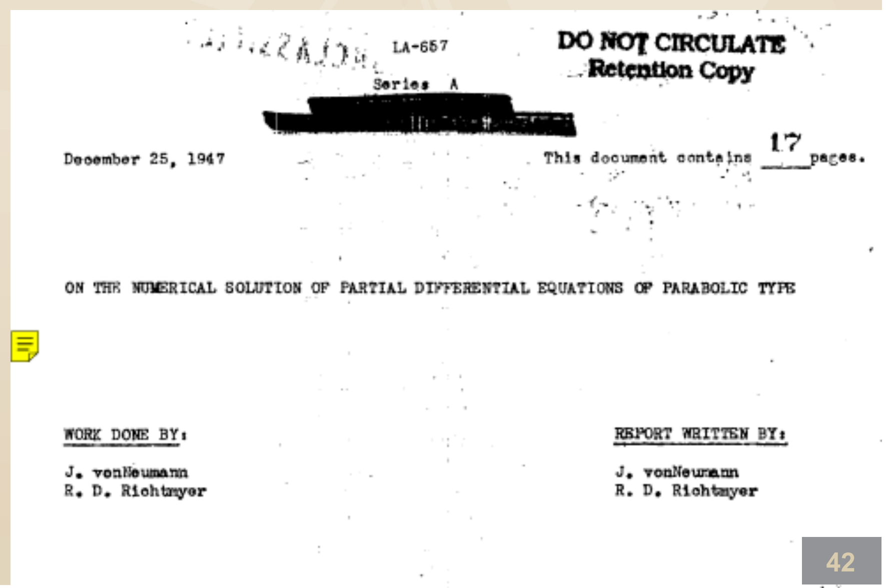
$$u_j^n = \exp\left(ij\theta\right) \Rightarrow u_j^n = \cos\left(j\theta\right) + i\sin\left(j\theta\right)$$

- It assumes that the function is periodic and the equation is linear.
- It is limited to linear equations.
- Despite these limitations, conducting this analysis is a virtual prerequisitite for the ultimate utility of a method to solve nonlinear systems of equations.

First a bit of history…



- Von Neumann introduced the Fourier technique at Los Alamos in 1946 in a lecture.
- *It was originally classified!*
- Used to analyze parabolic PDE integrators in 1947 LA Report (LA-657)
- Related to L_2 norm,…
  - …Energy norm
  - We'll do other norms, L_1



LA-657
Series A

DO NOT CIRCULATE
Retention Copy

[redacted]

December 25, 1947

This document contains 17 pages.

ON THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS OF PARABOLIC TYPE

WORK DONE BY:

J. vonNeumann
R. D. Richtmyer

REPORT WRITTEN BY:

J. vonNeumann
R. D. Richtmyer

Analysis of upwind differencing

- Substitute the Fourier series for the grid function

$$u_j^n = \exp(ij\theta) \Rightarrow u_j^{n+1} = u_j^n - c\left(u_j^n - u_{j-1}^n\right) \Rightarrow$$

$$A\exp(ij\theta) = \exp(ij\theta) - C\left(\exp(ij\theta) - \exp\left(i\left(j-1\right)\theta\right)\right)$$

- Expand into trigonometric functions and collect real and imaginary parts $A = 1 - C\left(1 - \exp(-i\theta)\right)$

$$A = 1 - C\left(1 - \cos(\theta) + i\sin(\theta)\right)$$

- Define the amplification and phase error

$$\mathbf{amp} = \sqrt{\left[1 - C\left(1 + \cos(\theta)\right)\right]^2 + \left(-C\sin(\theta)\right)^2}$$

$$\mathbf{phase} = \arctan\left(\frac{-C\sin(\theta)}{\left[1 - C\left(1 + \cos(\theta)\right)\right]}\right) \Big/ \left(-c\theta\right)$$

Where does this phase (wave dispersion) error formula come from?

- Write the numerical scheme as

$$u^{n+1} = S(\theta)u^n \rightarrow u^{n+1} = \left|S(\theta)\right|\exp\left(-i\alpha(\theta)\right)u^n$$

- The correct answer is

$$u^{n+1} = u^n \exp\left(-iC\theta\right)$$

- The argument (phase angle) of the numerical scheme can be found and compared to the exact solution

$$\operatorname{Arg}\left(S(\theta)\right) = \arctan\left[\frac{\operatorname{Im}\left(S(\theta)\right)}{\operatorname{Re}\left(S(\theta)\right)}\right]$$

- The same for the exact solution gives $-C\theta$

Analysis of upwind differencing (continued)

- Perform an asymptotic expansion in small angles
  - Amplitude error even order errors

$$\mathbf{amp} \approx 1 + \left(-\frac{c}{2} + \frac{c^2}{2} \right) \theta^2 + O\left(\theta^4\right)$$

  - Phase error odd order (divide by the angle!)

$$\mathbf{phase} \approx 1 + \left(-\frac{1}{6} + \frac{c}{2} - \frac{c^2}{3} \right) \theta^2 + O\left(\theta^4\right)$$

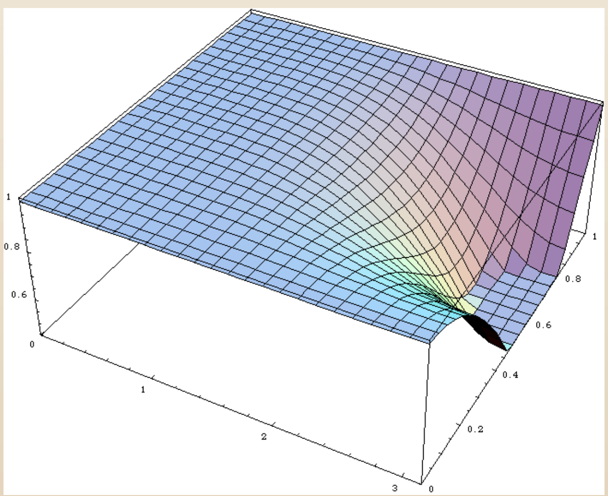
- Bound the function for all angles and find the CFL limit (error goes to zero at CFL=1, then unstable).

Looking at error one must remember that this is for a single time step.

- If one wants to get to the amount of error for a fixed period of time, the formulas need to change.
- A fixed point in time will take

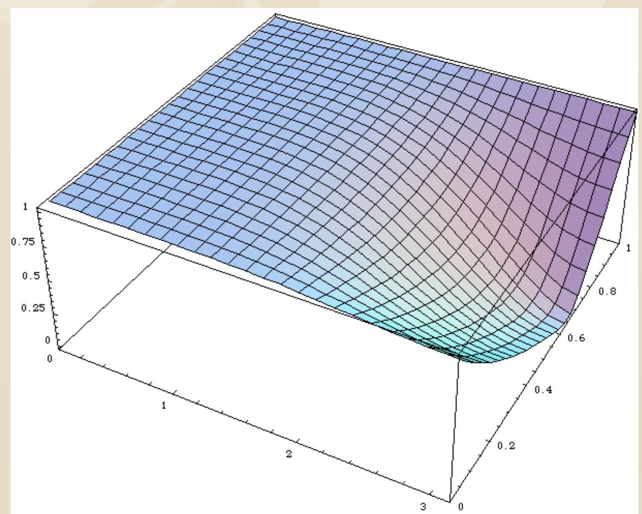
$$n = \frac{1}{c}$$

- For example looking at amplitude error



$$\left|S(c,\theta)\right| \longleftarrow$$

$$\left|S(c,\theta)\right|^{1/c} \longrightarrow$$



There is a closely related analysis of only the time-differencing (semi-discrete)

- One can derive the ideal behavior by taking the derivative of the spatial gradient,

$$\frac{\partial u}{\partial x} \Rightarrow \frac{\partial u}{\partial \theta} = i \exp\left(i\theta\right)$$

- This form has found great use in designed high order methods
  - Ideal schemes have no dissipation and only dispersion error.

Let’s look at an example of 2nd order centered differencing versus upwinding.

- The symbol for 2nd order centered is

$$i\sin(\theta)$$

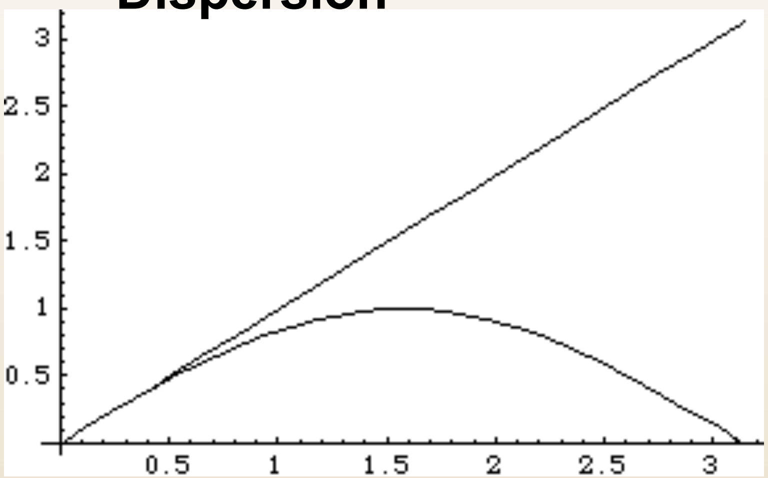
- The symbol for 1st order upwind is

$$1-\cos(\theta)+i\sin(\theta)$$

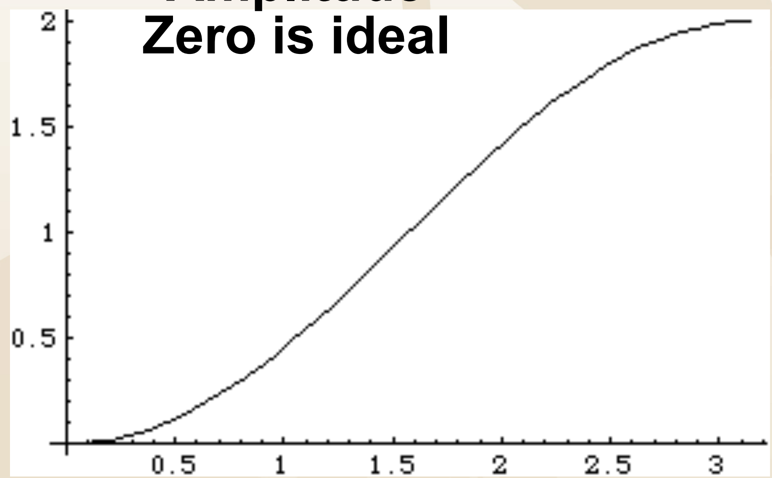
  - The real part represents dissipation
  - And the imaginary part is the ideal operator (and the same as second-order centered)

Analyzed by plotting the parts of the symbol versus the ideal - for upwinding

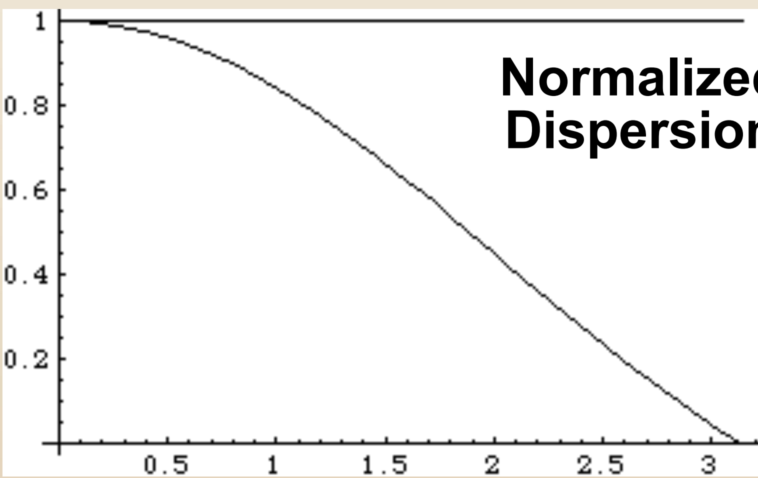
Dispersion



Amplitude
Zero is ideal



Normalized Dispersion



The reason for doing modified equation analysis isn't always clear.

"…the progress of physics will to a large extent depend on the progress of nonlinear mathematics, of methods to solve nonlinear equations. …and therefore we can learn by comparing different nonlinear problems." – Werner Heisenberg

The technique for modified equation analysis was introduced by Hirt.

- Hirt (1968) introduced the technique and examined the truncation errors in physical terms.
- Warming and Hyett (1974) discussed the method in great detail and provided an analysis framework for fully discrete integrators.

JOURNAL OF COMPUTATIONAL PHYSICS 2, 339–355 (1968)

Heuristic Stability Theory for Finite-Difference Equations*

C. W. HIRT

Los Alamos Scientific Laboratory, University of California, Los Alamos, New Mexico 87544

JOURNAL OF COMPUTATIONAL PHYSICS 14, 159–179 (1974)

The Modified Equation Approach to the Stability and Accuracy Analysis of Finite-Difference Methods

R. F. WARMING AND B. J. HYETT

Computational Fluid Dynamics Branch,
Ames Research Center, NASA, Moffett Field, California 94035

Received June 11, 1973

The modified equation technique is an important augmentation to Fourier analysis.

- The key to modified equation analysis (MEA) is the ability to..
  - …see the errors in differential form,...
  - …and extend the analysis to include nonlinearity.
- This gives us several advantages:
  - The truncation errors can be studied in terms of differential equations and directly compared with physical or modeled terms,
  - and directly treat nonlinear physics or numerics.

Semi-discrete is much simplier to deal with!

- None of this time-space cancelation is necessary.
- Consider a fourth-order space difference

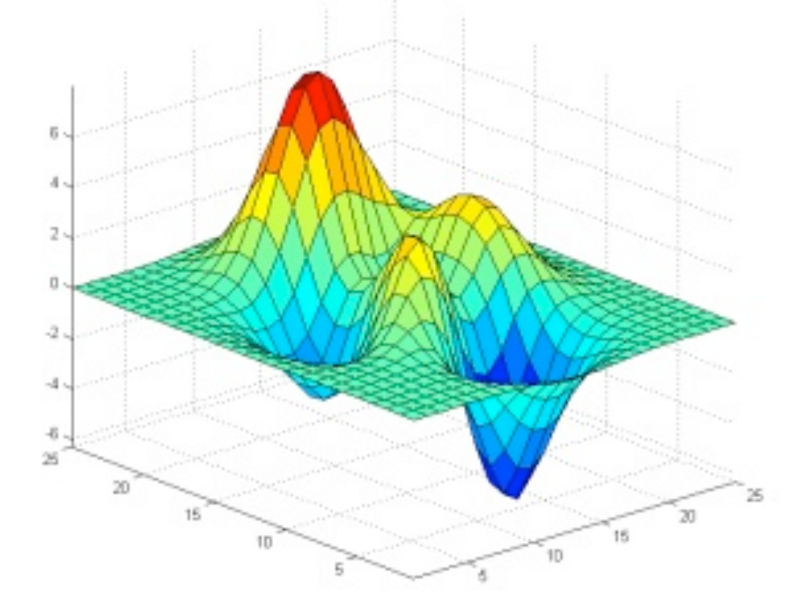
$$\frac{\partial u}{\partial x} \approx \frac{2}{3\Delta x}\left(u_{j+1} - u_{j-1}\right) - \frac{1}{12\Delta x}\left(u_{j+2} - u_{j-2}\right)$$

- Take the Taylor expansion,

$$\frac{\partial u}{\partial x} - \frac{\Delta x^4}{30}\frac{\partial^5 u}{\partial x^5} - \frac{\Delta x^6}{252}\frac{\partial^7 u}{\partial x^7}$$

What proof (verification) means in numerical analysis!

"For the numerical analyst there are two kinds of truth; the truth you can prove ***and the truth you see when you compute***." – Ami Harten



Another thought courtesy of Peter Lax

"How fortunate that in this best of all possible worlds the equations of ideal flow are nonlinear!"

SIAM Review, January 1969

Of course how could it be otherwise?

TURBULENCE MODELING

Some quotes to set our thoughts in the right direction.

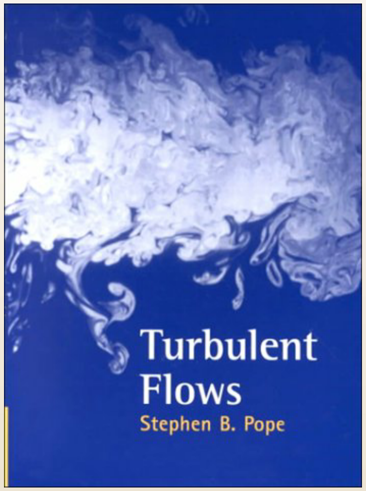
- “I am an old man now, and when I die and go to heaven there are two matters on which I hope for enlightenment. One is quantum electrodynamics, and the other is the turbulent motion of fluids. And about the former I am rather optimistic.” - Horace Lamb -

- “Turbulence is the most important unsolved problem of classical physics.” - Richard Feynman

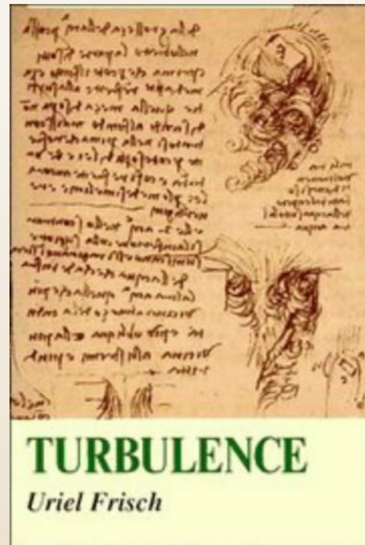
References for this Lecture

Lots of good books on turbulence!

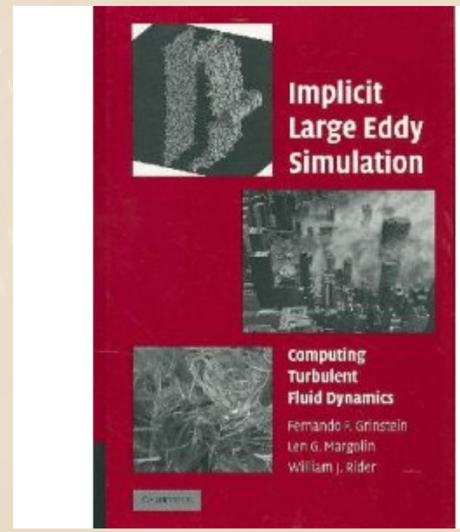
Pope



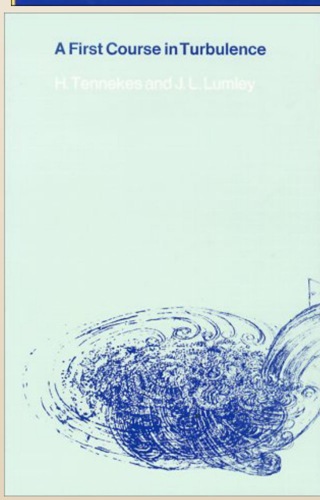
Frisch



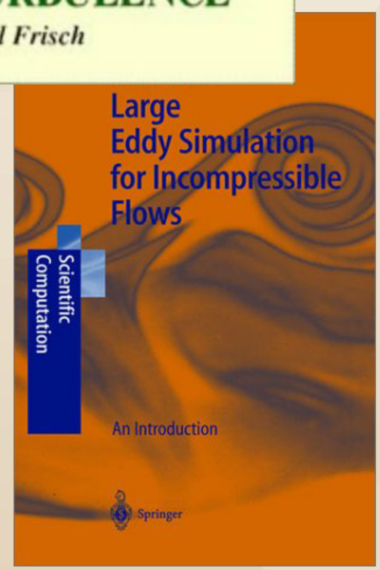
My own as an editor



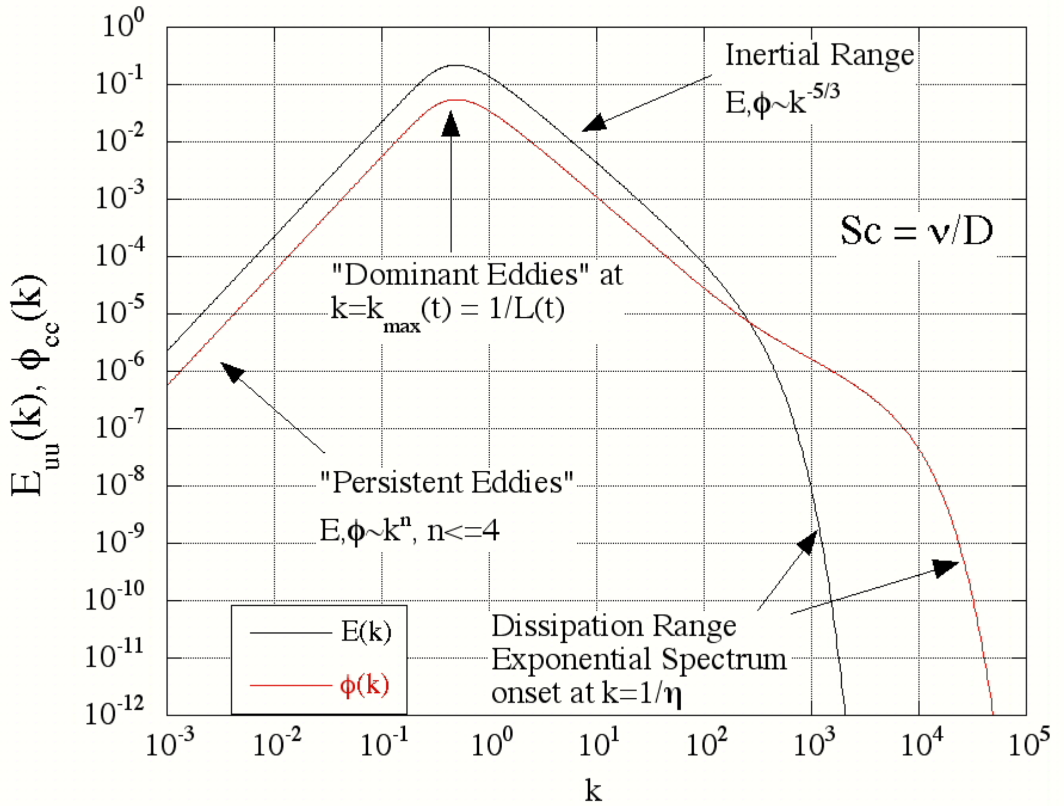
Lumley



Saugaut



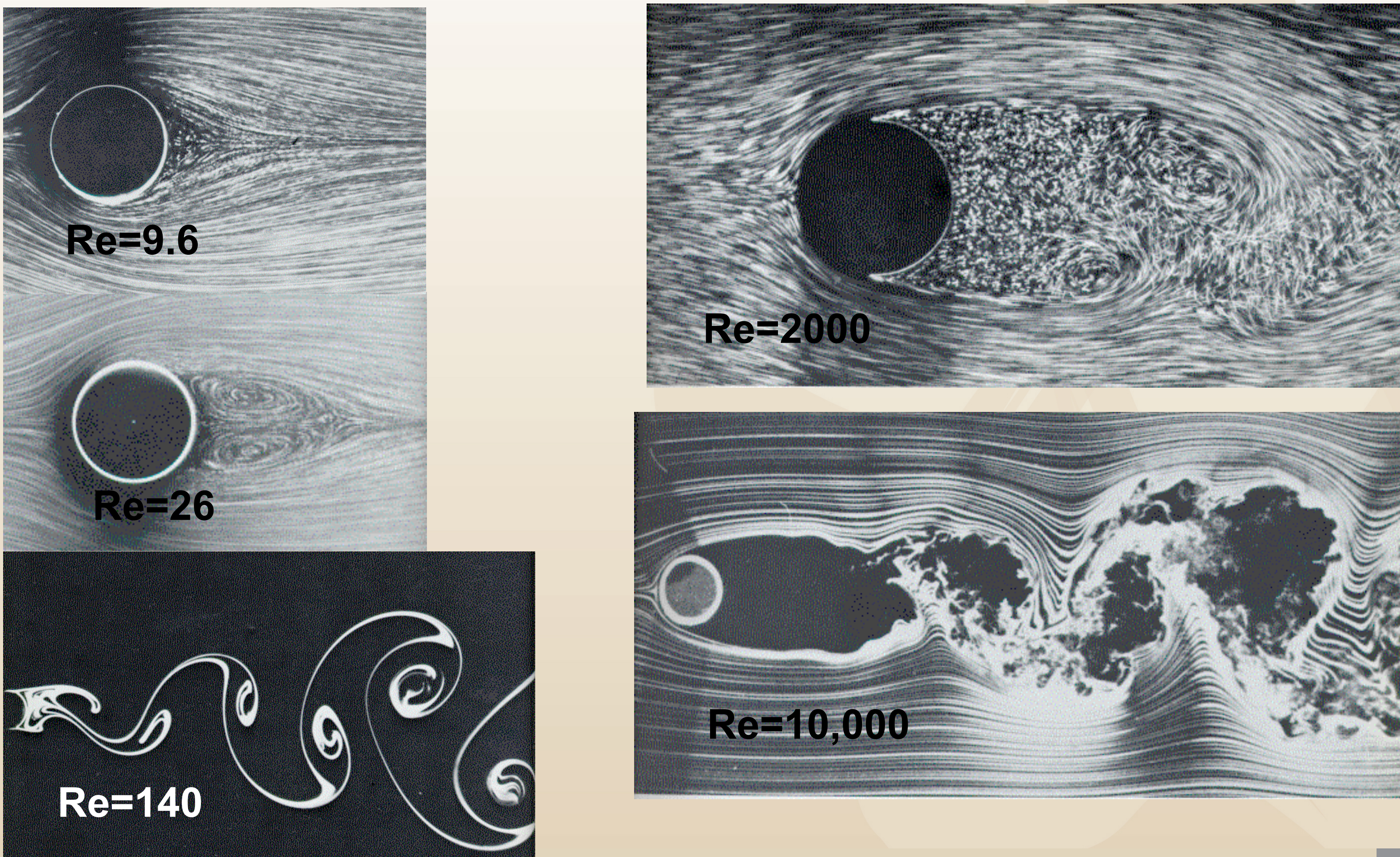
There is some basic turbulent theory.



Energy and Passive Scalar Spectra for Large Schmidt Number
CMSB Model, Rubinstein & Clark, 2003

- There are the large scales where energy enters the flow.
- The inertial range where energy flows to smaller scales.
- The dissipation range where the energy goes to heat.
- Material mixing will takes place at a different scale.

Here is a simple explanation of turbulence.



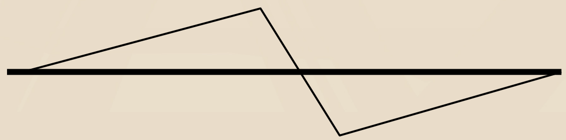
There is some basic turbulent theory.

- Turbulence is best understood through a simple abstraction using symmetry, isotropy flows that are homogeneous.
- The smallest scale is the Kolmogorov scale where energy is dissipated,

$$\eta = \left(\frac{\langle \dot{K} \rangle}{\nu} \right)^{1/3}$$

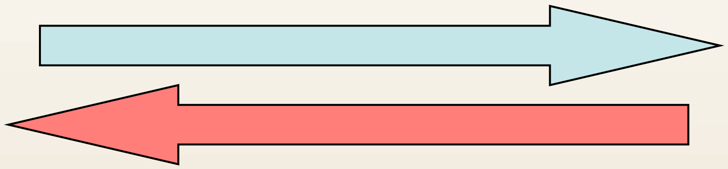
- The inertial range is defined by an analytical result (Kolmogorov's 4/5 law)

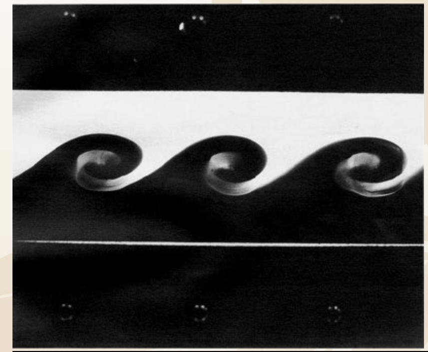
$$\frac{4}{5} \langle \dot{K} \rangle \ell = \left\langle \left[u\left(x + \ell \cdot \hat{n}\right) - u\left(x\right) \right]^3 \right\rangle$$



Turbulence is produced by instabilities.

- Kelvin-Helmholtz

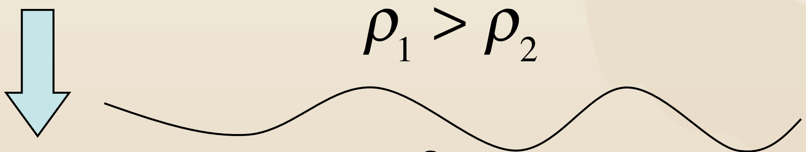




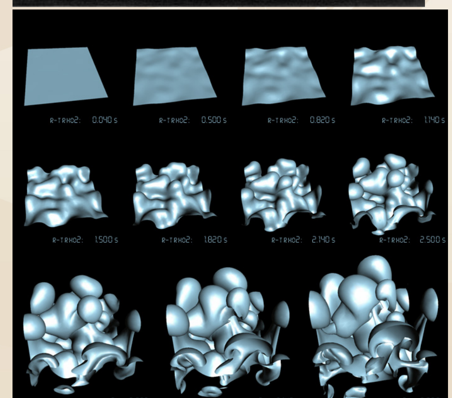
- Rayleigh-Taylor

gravity

$\rho_1 > \rho_2$

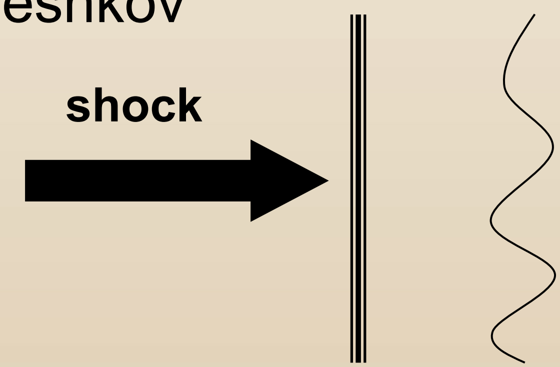


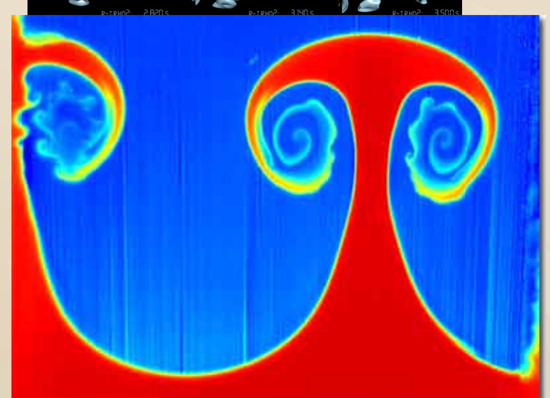
ρ_2



- Richtmyer-Meshkov

shock





Turbulence modeling falls into several distinct categories.

- The simplest thing to do is to apply an eddy viscosity, mixing length theory to compute the mean dissipation due to the presence of turbulence..
- The eddy viscosity can be defined by equations, kinetic energy, or two equations, or more. This is usually associated with Reynolds Averaged Navier-Stokes (RANS).
- Large eddy simulation (LES) where the key is subgrid modeling and numerical resolution.
- Implicit large eddy simulation (ILES) avoids subgrid modeling, but focuses on the numerical methodology.
- Finally, direct numerical simulation (DNS) where the complete turbulent physics are computed without models.

The simplest models are called Reynolds Averaged Navier-Stokes (RANS) models

- One can then define equations for production and dissipation of turbulent kinetic energy, and then perhaps their moments for even more equations.
  - One can consider this a sequence of moments
  - 1st moment is the KE, the 2nd moment is the variation in KE are the most complex models currently.
- One of the key aspects of this approach is the averaging through defining an ensemble, a sequence of events that is averaged over.
- The model is supposed to give the results for the ensemble of such flows.

Closures are often based on a eddy viscosity, that graduates to evolution equations for eddy viscosity.

- Simplest is eddy viscosity, ala Prandtl

$$\frac{\partial u_i u_j}{\partial x_i} = \frac{\partial\left(\bar{u}_i + u_i'\right)\left(\bar{u}_j + u_j'\right)}{\partial x_i} = \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} + \frac{\partial u_i' u_j'}{\partial x_i} \approx \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} + \frac{\partial}{\partial x_i} \nu_T \frac{\partial \bar{u}_i}{\partial x_i}$$

$$\nu_T = \ell^2 \left| \frac{\partial \bar{u}_i}{\partial x_i} \right|$$

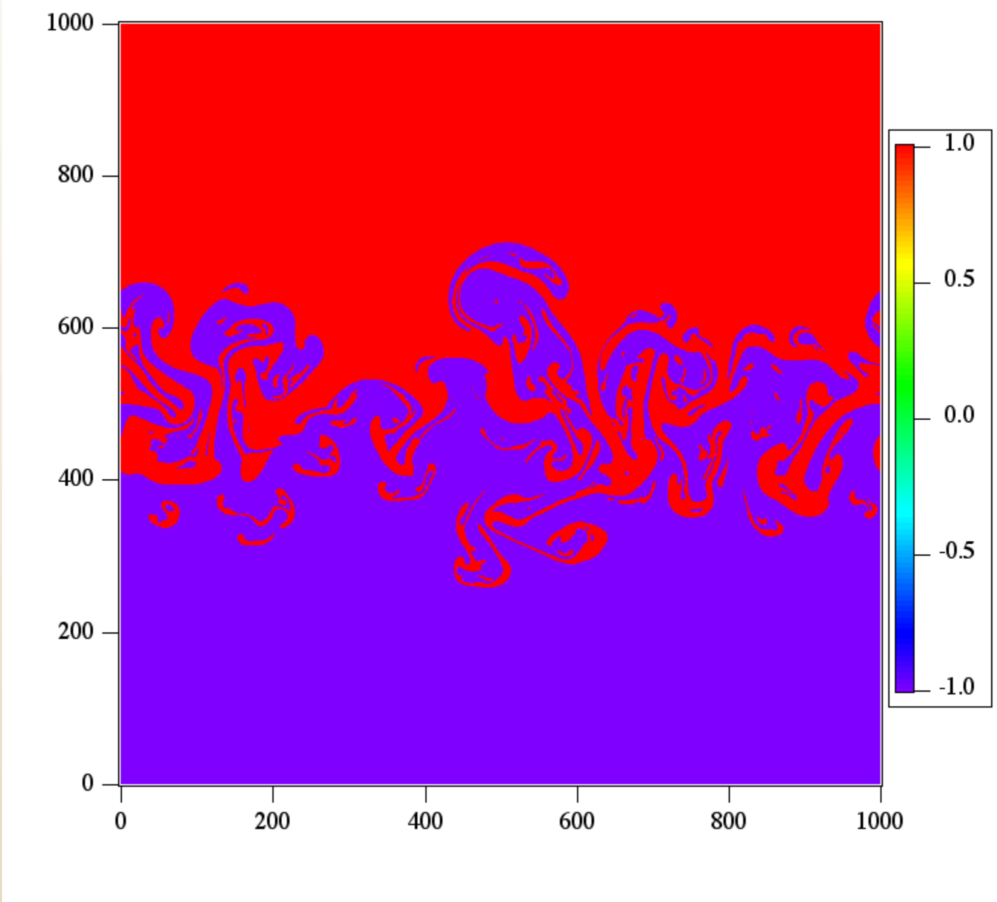
- Next, an evolution equation is defined for the turbulent kinetic energy

$$\frac{Dk}{Dt} = \nabla \cdot \frac{\nu_T}{\sigma} \nabla k + \mathcal{P} - \varepsilon$$

  – With an eddy viscosity defined by $\nu_T = c\ell\sqrt{k}$

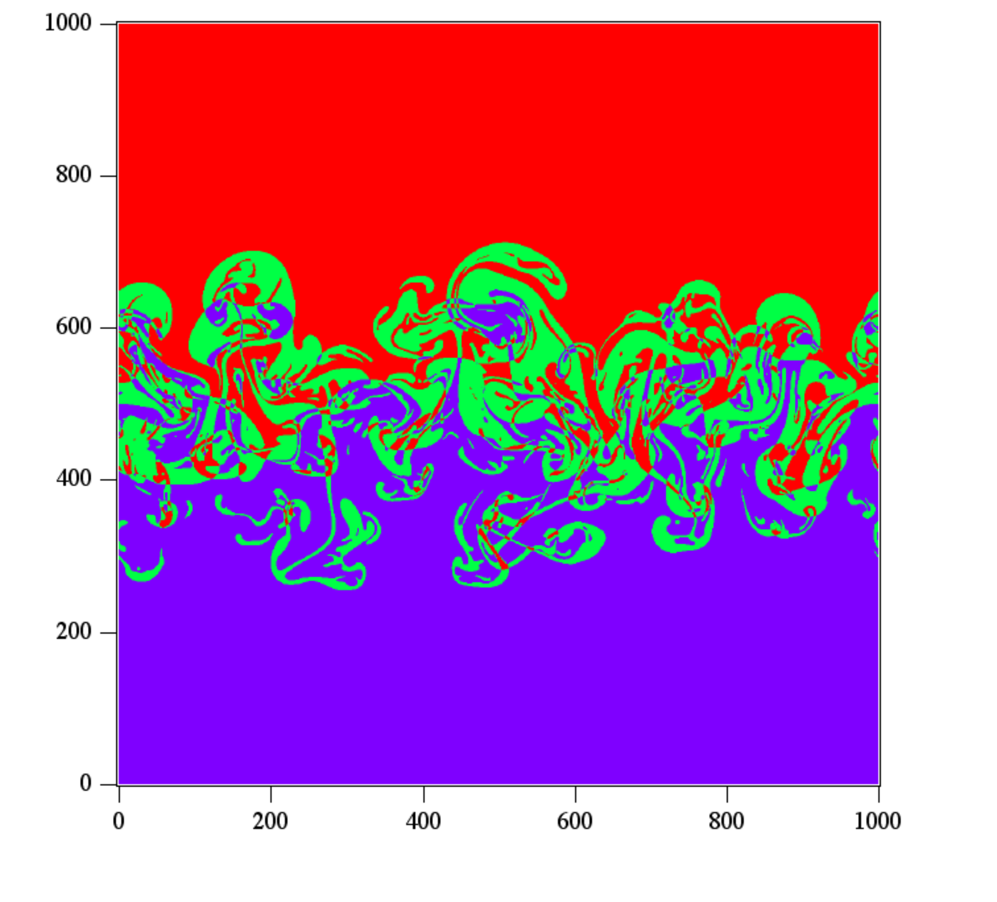
An example from Tim Clark (formerly of LANL) to illustrate the differences between LES and RANS.

- Simulation is two-dimensional, immiscible (no numerical diffusion). Size = 1000x1000
- Figures sequentially shows an average of 1, 2, 5, 10, 25 and 75 realizations.
- Each realization started as a quiescent flow.
- Perturbation spectrum and RMS value of perturbations are equal for all realizations.
- The result of ensemble averaging appears “diffusive” but is not! The distinction between a diffusive versus a structured mix is determined by the covariance of density or concentration–information included in a RANS model but not in an LES.



One Realization

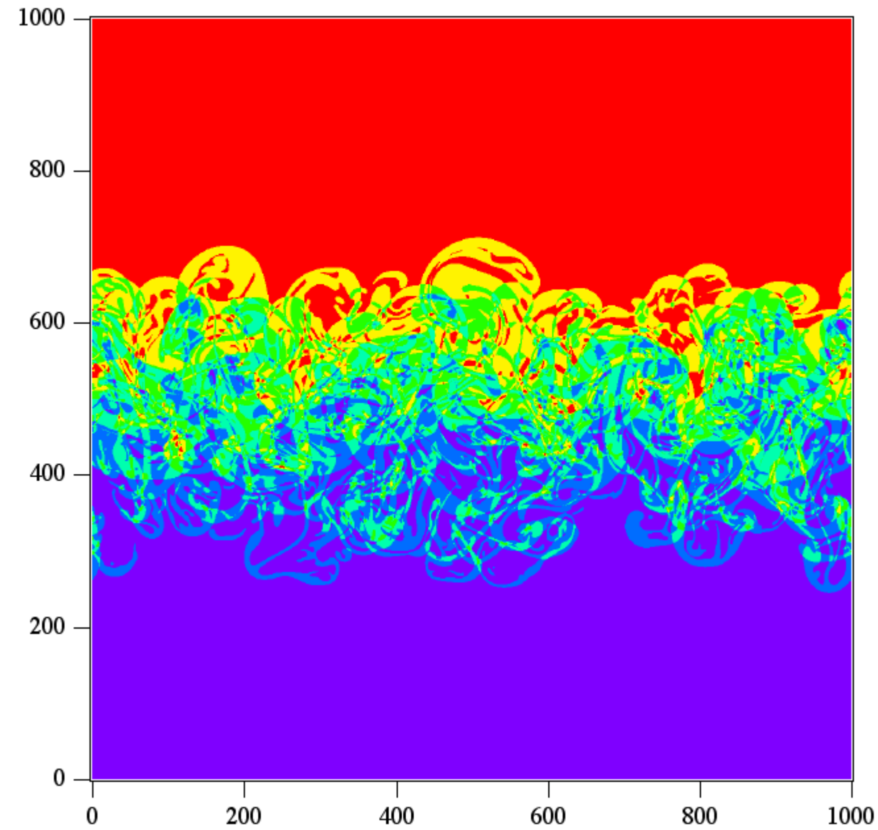
Example of an Exact Ensemble Average



Two Realizations

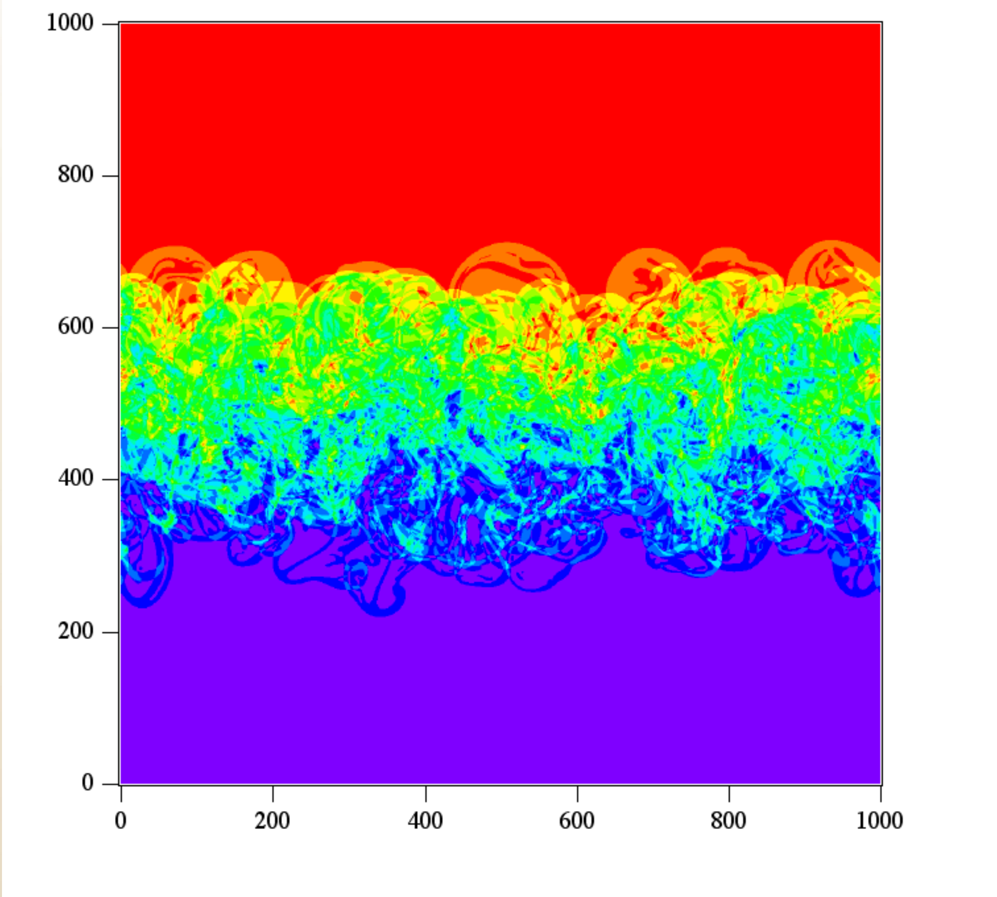
Example of an Exact Ensemble Average

- Simulation is two-dimensional, immiscible (no numerical diffusion). Size = 1000x1000
- Figures sequentially shows an average of 1, 2, 5, 10, 25 and 75 realizations.
- Each realization started as a quiescent flow.
- Perturbation spectrum and RMS value of perturbations are equal for all realizations.
- The result of ensemble averaging appears "diffusive" but is not! The distinction between a diffusive versus a structured mix is determined by the covariance of density or concentration–information included in a RANS model but not in an LES.



Five Realizations

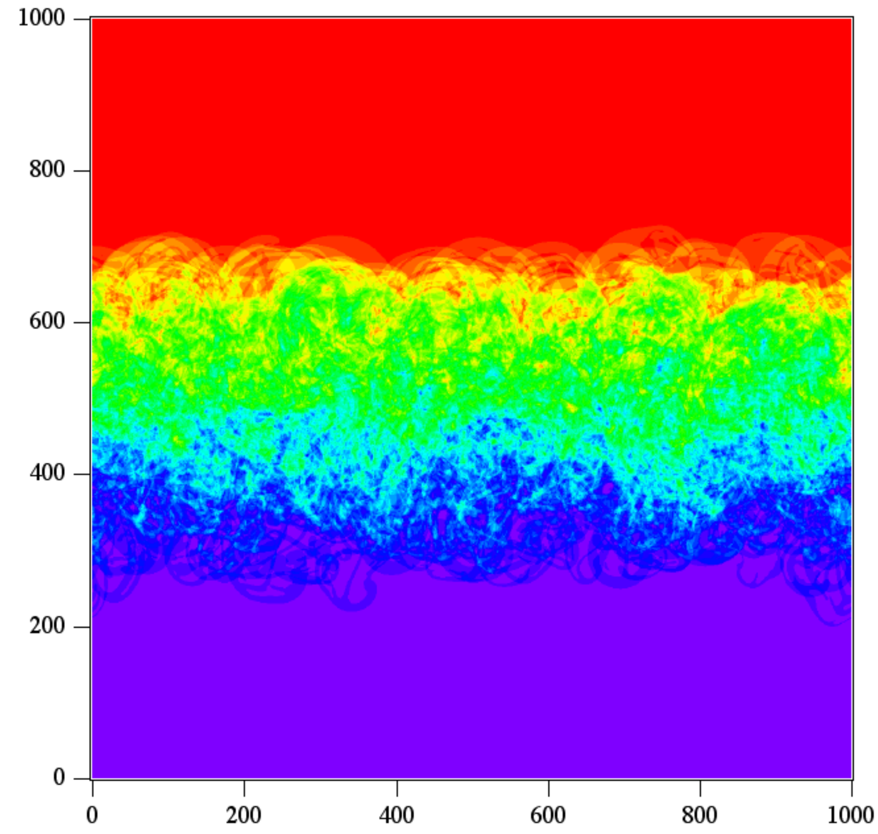
Example of an Exact Ensemble Average



Ten Realizations

Example of an Exact Ensemble Average

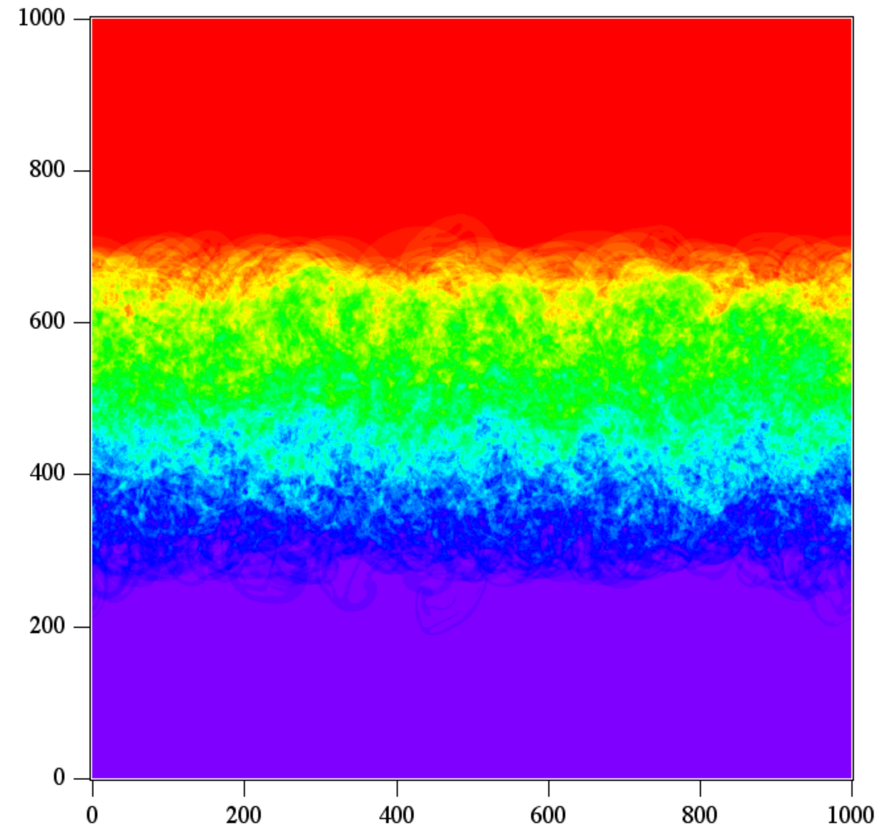
- Simulation is two-dimensional, immiscible (no numerical diffusion). Size = 1000x1000
- Figures sequentially shows an average of 1, 2, 5, 10, 25 and 75 realizations.
- Each realization started as a quiescent flow.
- Perturbation spectrum and RMS value of perturbations are equal for all realizations.
- The result of ensemble averaging appears “diffusive” but is not! The distinction between a diffusive versus a structured mix is determined by the covariance of density or concentration–information included in a RANS model but not in an LES.



Twenty Five Realizations

Example of an Exact Ensemble Average

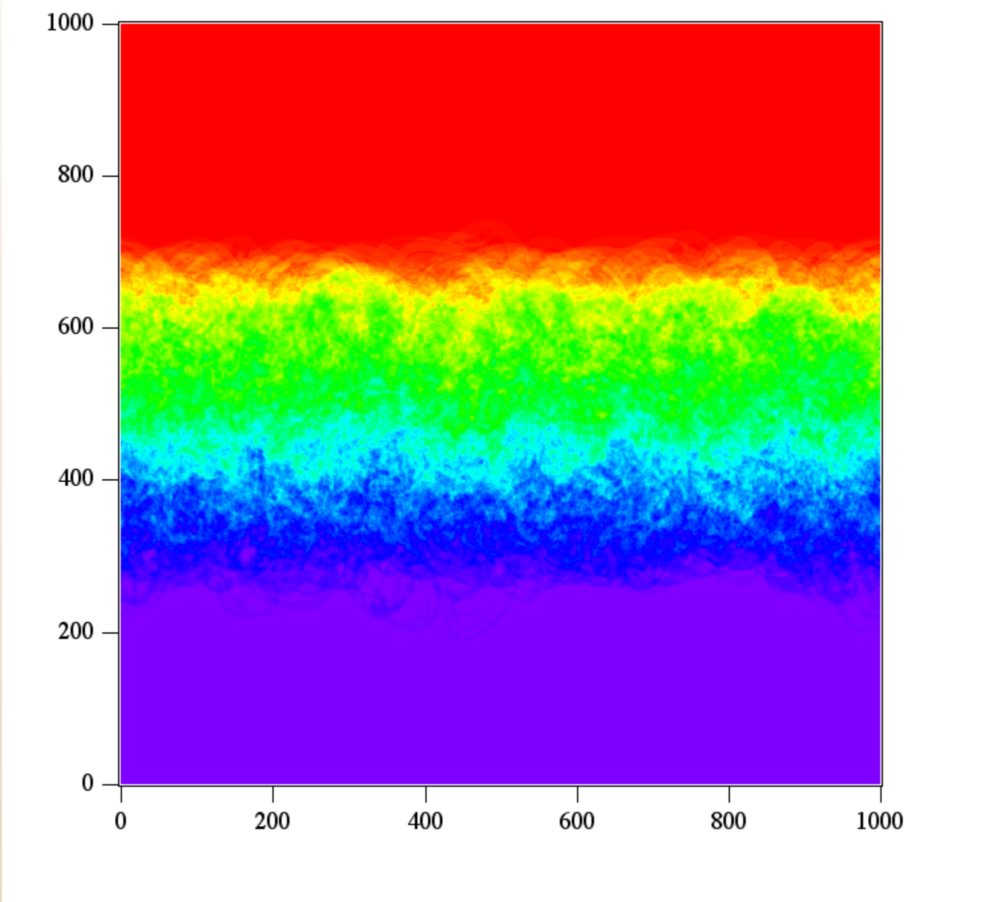
- Simulation is two-dimensional, immiscible (no numerical diffusion). Size = 1000x1000
- Figures sequentially shows an average of 1, 2, 5, 10, 25 and 75 realizations.
- Each realization started as a quiescent flow.
- Perturbation spectrum and RMS value of perturbations are equal for all realizations.
- The result of ensemble averaging appears “diffusive” but is not! The distinction between a diffusive versus a structured mix is determined by the covariance of density or concentration–information included in a RANS model but not in an LES.



Fifty Realizations

Example of an Exact Ensemble Average

- Simulation is two-dimensional, immiscible (no numerical diffusion). Size = 1000x1000
- Figures sequentially shows an average of 1, 2, 5, 10, 25 and 75 realizations.
- Each realization started as a quiescent flow.
- Perturbation spectrum and RMS value of perturbations are equal for all realizations.
- The result of ensemble averaging appears “diffusive” but is not! The distinction between a diffusive versus a structured mix is determined by the covariance of density or concentration–information included in a RANS model but not in an LES.



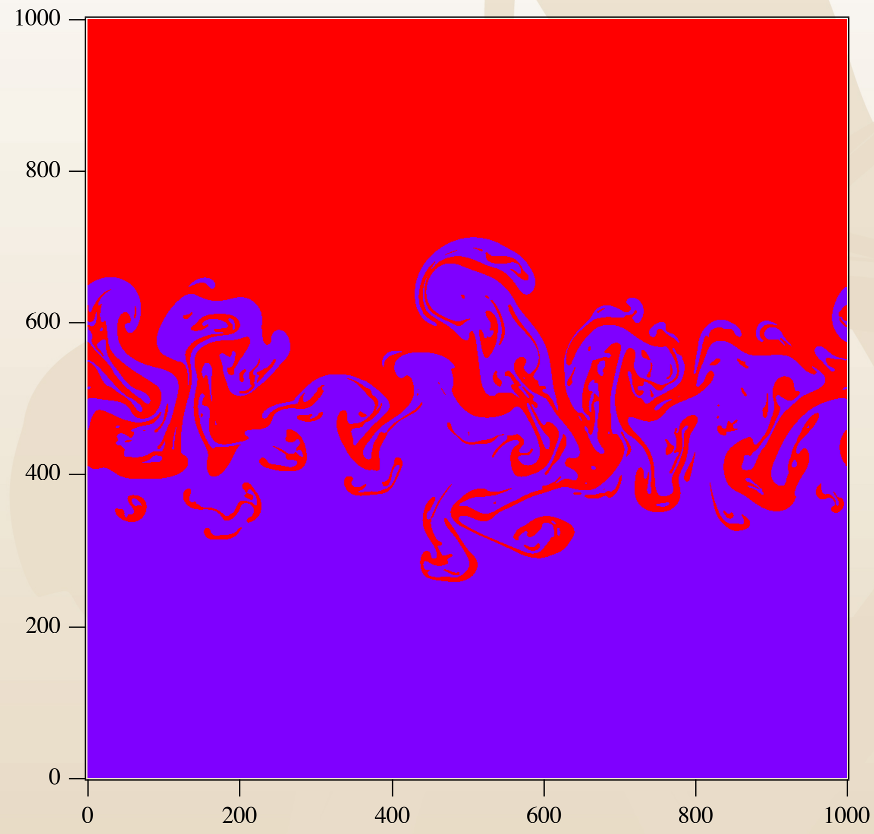
Seventy Five Realizations

Next, we have large eddy simulation.

- Perhaps the philosophy of LES comes down to this,
  - "No amount of genius can overcome a preoccupation with detail" - Levy's Eighth Law

Example of an Exact Spatial Average: Again from Tim Clark

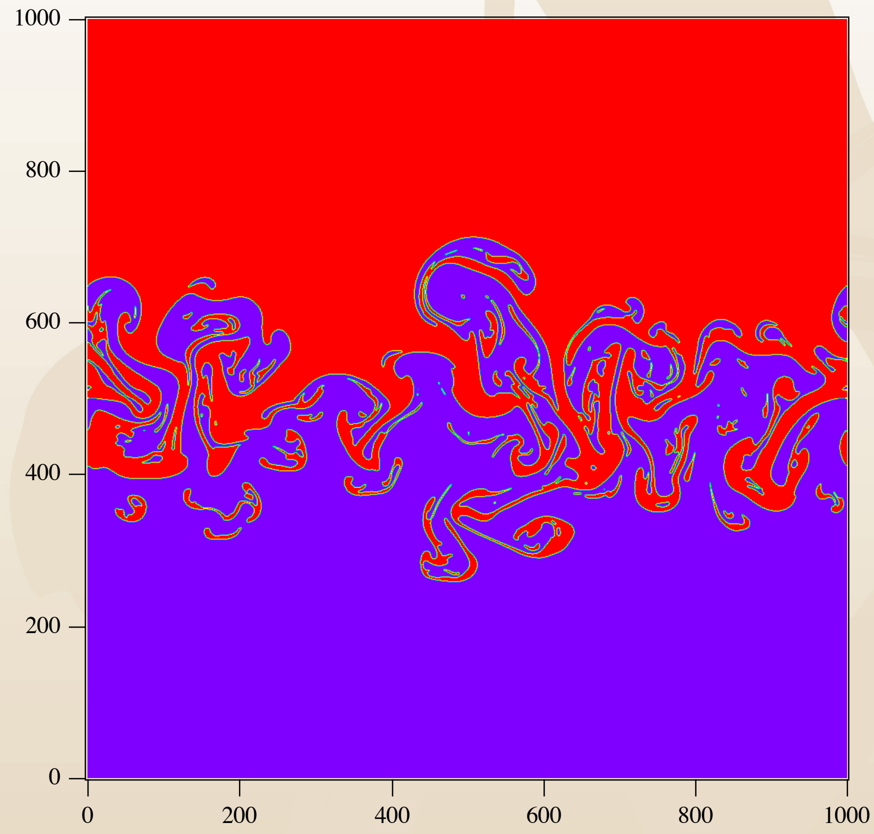
- Same simulations as used for ensemble example.
- Spatial averages are over a “circular” region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: “smeared” result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be “optimistic”–the obvious limit for larger filter sizes is a featureless smear.



Radius = 0.5 “Cells” (Unaveraged)

Example of an Exact Spatial Average

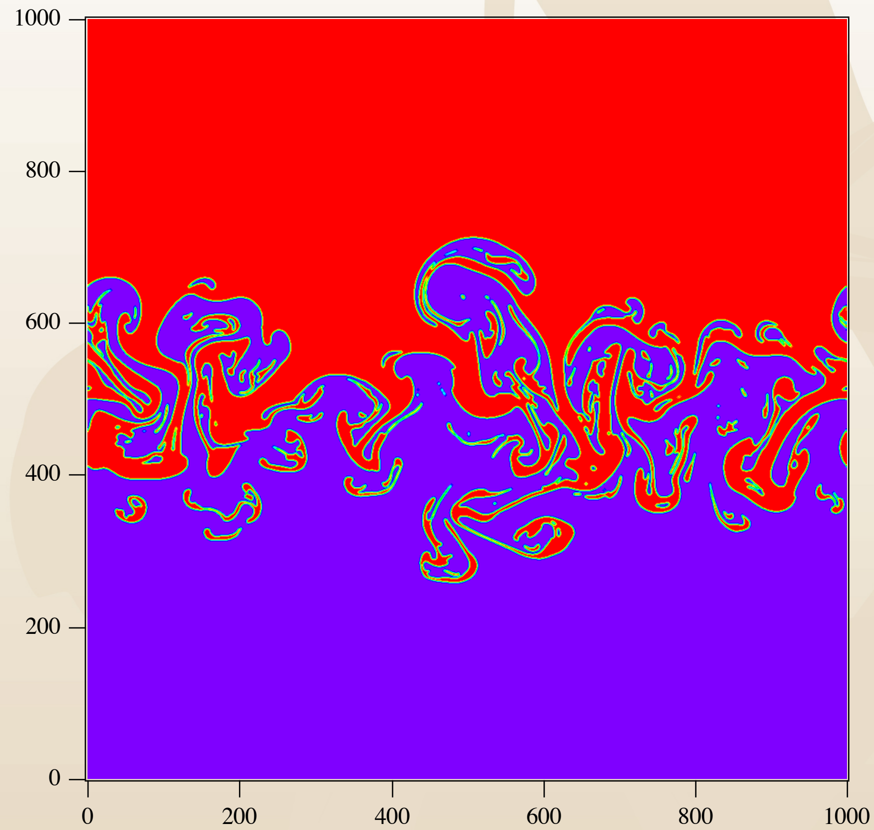
- Same simulations as used for ensemble example.
- Spatial averages are over a "circular" region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: "smeared" result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be "optimistic"–the obvious limit for larger filter sizes is a featureless smear.



Radius = 1.5 "Cells"

Example of an Exact Spatial Average

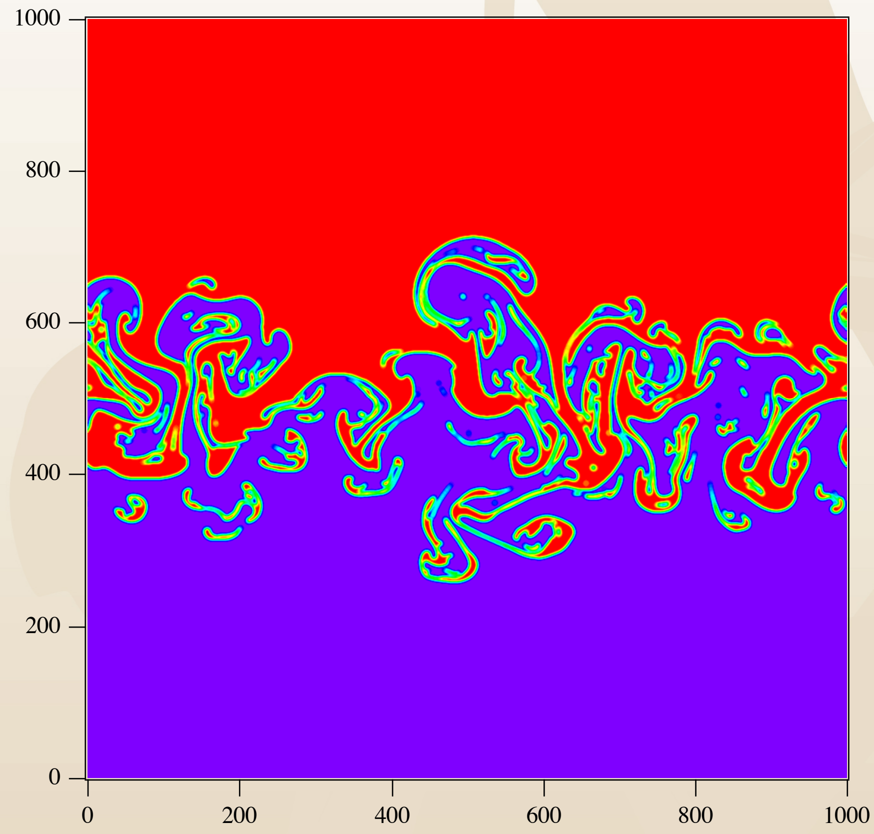
- Same simulations as used for ensemble example.
- Spatial averages are over a “circular” region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: “smeared” result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be “optimistic”–the obvious limit for larger filter sizes is a featureless smear.



Radius = 2.5 “Cells”

Example of an Exact Spatial Average

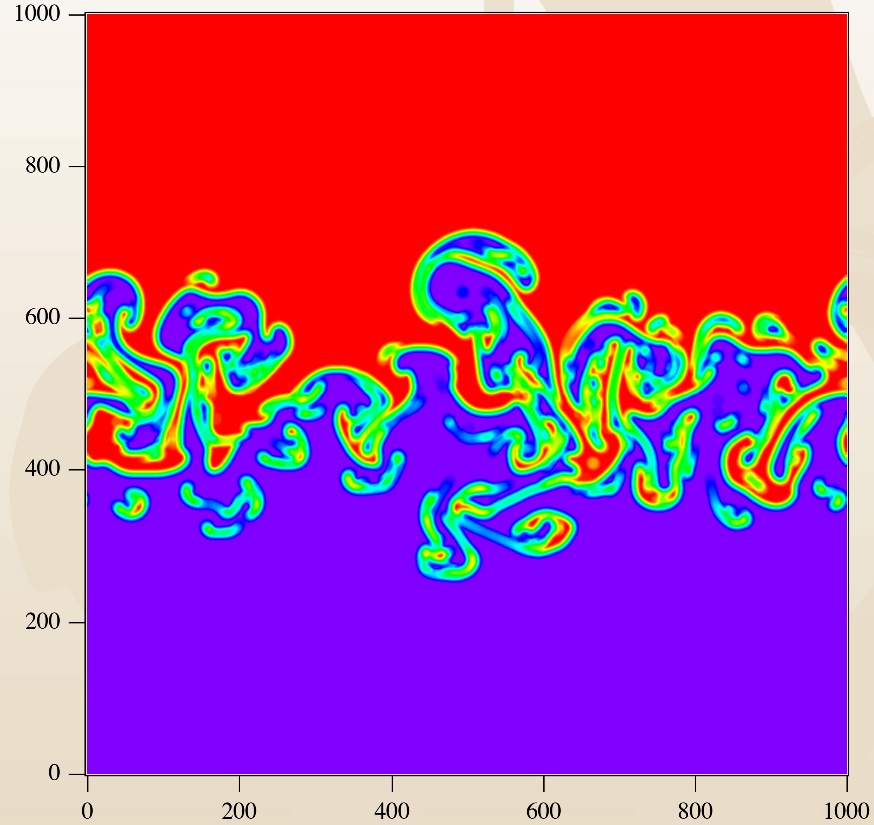
- Same simulations as used for ensemble example.
- Spatial averages are over a “circular” region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: “smeared” result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be “optimistic”–the obvious limit for larger filter sizes is a featureless smear.



Radius = 4.5 “Cells”

Example of an Exact Spatial Average

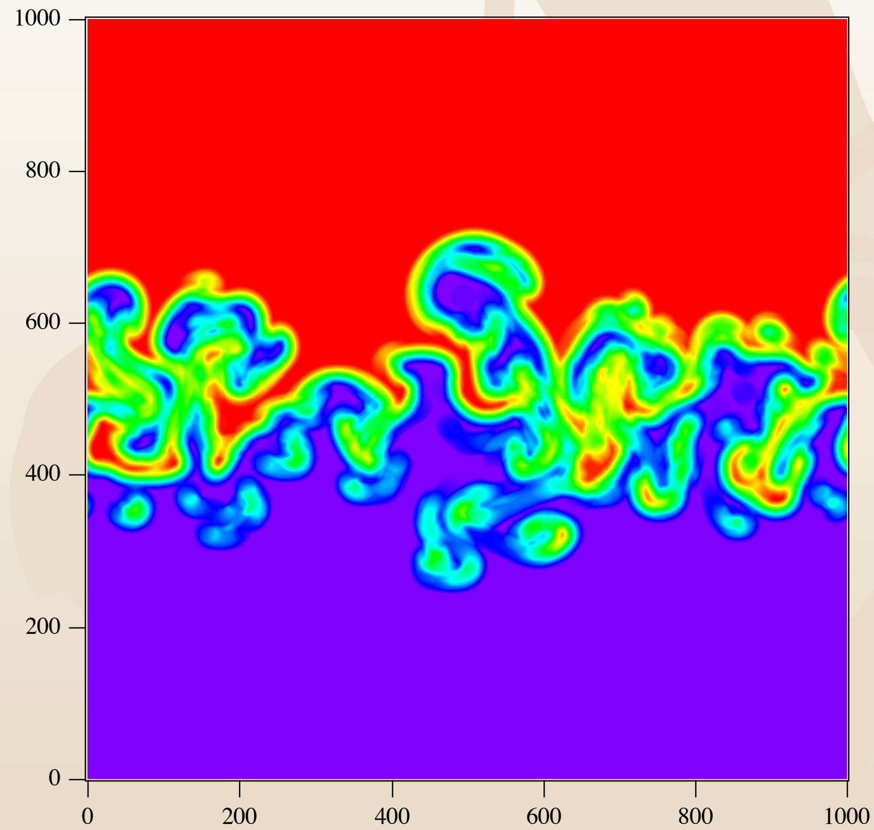
- Same simulations as used for ensemble example.
- Spatial averages are over a "circular" region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: "smeared" result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be "optimistic"–the obvious limit for larger filter sizes is a featureless smear.



Radius = 8.5 "Cells"

Example of an Exact Spatial Average

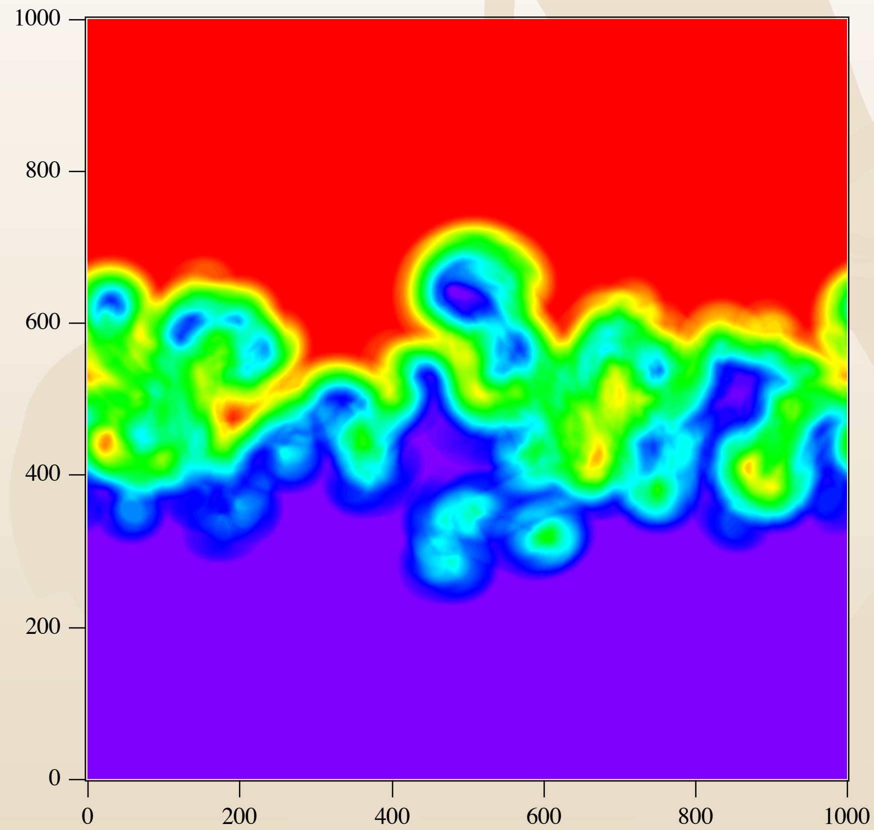
- Same simulations as used for ensemble example.
- Spatial averages are over a “circular” region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: “smeared” result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be “optimistic”–the obvious limit for larger filter sizes is a featureless smear.



Radius = 16.5 “Cells”

Example of an Exact Spatial Average

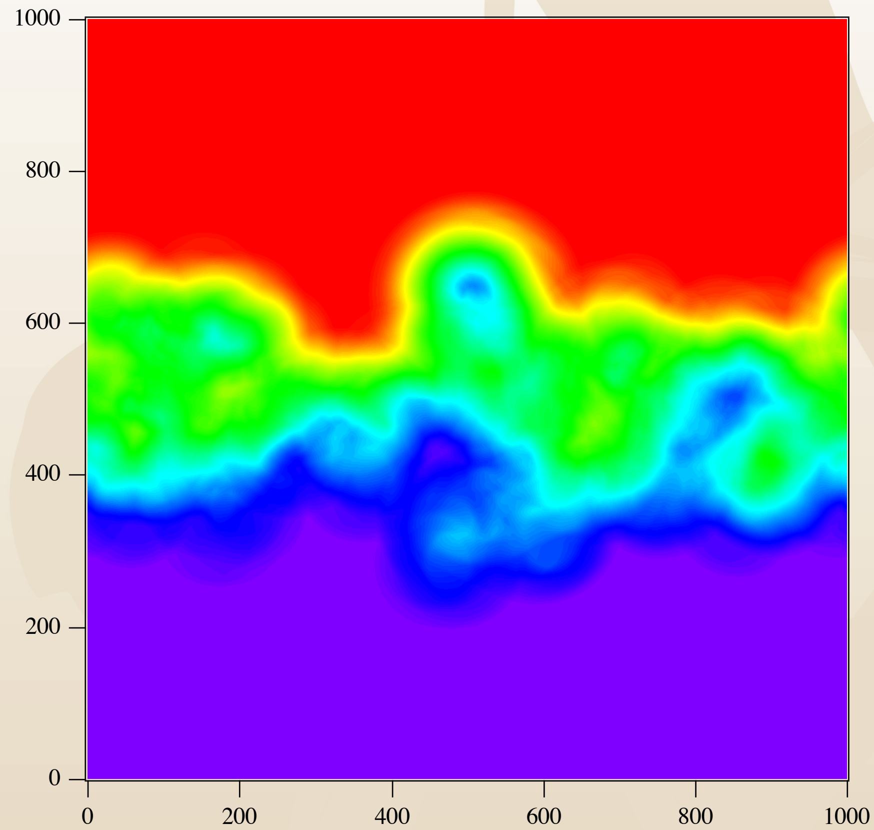
- Same simulations as used for ensemble example.
- Spatial averages are over a "circular" region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: "smeared" result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be "optimistic"–the obvious limit for larger filter sizes is a featureless smear.



Radius = 32.5 "Cells"

Example of an Exact Spatial Average

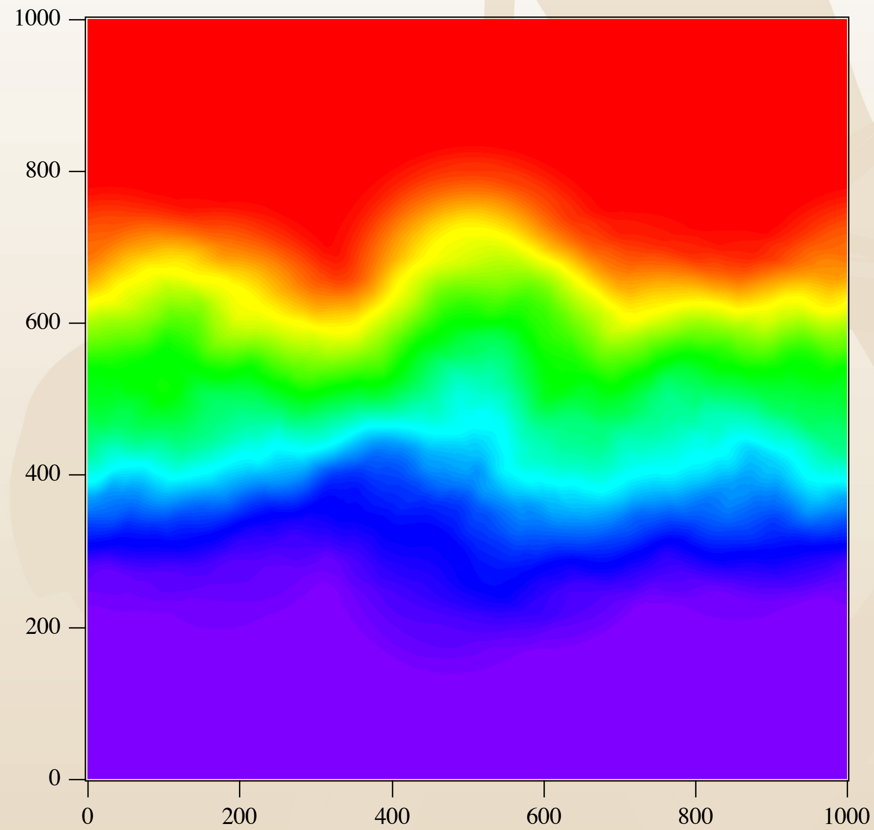
- Same simulations as used for ensemble example.
- Spatial averages are over a "circular" region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: "smeared" result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be "optimistic"–the obvious limit for larger filter sizes is a featureless smear.



Radius = 64.5 "Cells"

Example of an Exact Spatial Average

- Same simulations as used for ensemble example.
- Spatial averages are over a "circular" region with radii of (sequentially) 1, 2, 4, 8, 16, 32 and 64 cells.
- Fine detail and two-phase nature lost due to averaging: "smeared" result.
- LES methods use a spatial average convolved with a filter. They do not keep higher moments, thus the smearing is indistinguishable from molecular mixing.
- For practical calculations the 64 cell case may be "optimistic"–the obvious limit for larger filter sizes is a featureless smear.



Radius = 128.5 "Cells"

Large eddy simulation requires the numerical solution of part of the inertial range.

- The modeling is to provide the effect of the length scales truncated from the flow.
- The most famous model is the Smagorinsky model, an eddy viscosity model

$$\nabla \cdot \tau, \tau = C_{\mathbf{Smag}} h^2 \left\| \nabla u \right\| \nabla u$$

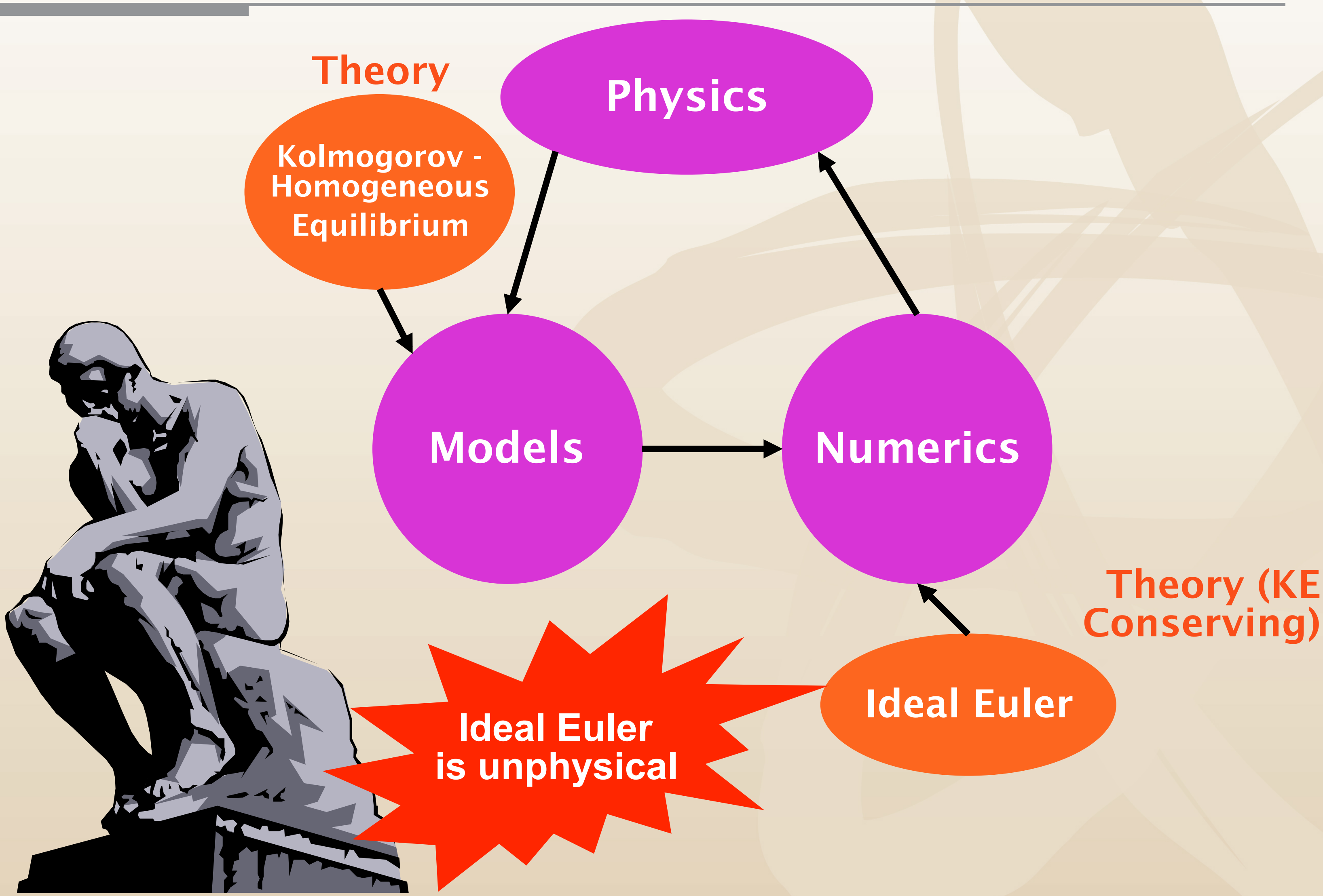
- Other models include non-dissipative effects (self-similarity)

$$\nabla \cdot \tau, \tau = C h^2 \nabla u \nabla u$$

The Smagorinsky model is based on Von Neumann-Richtmyer artificial viscosity.

- It was the suggestion of Jules Charney that Von Neumann's viscosity be used in modeling weather in order to stabilize the calculations (1956).
- Smagorinsky developed the 3-D version of artificial viscosity based on this suggestion.
- This was the start of LES modeling.

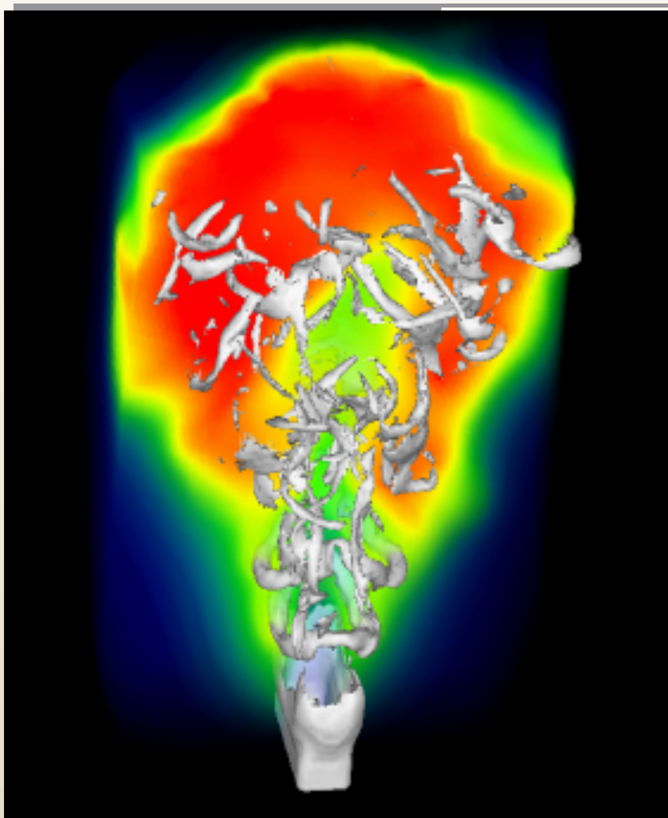
The standard modeling approach for turbulence is based on certain assumptions.



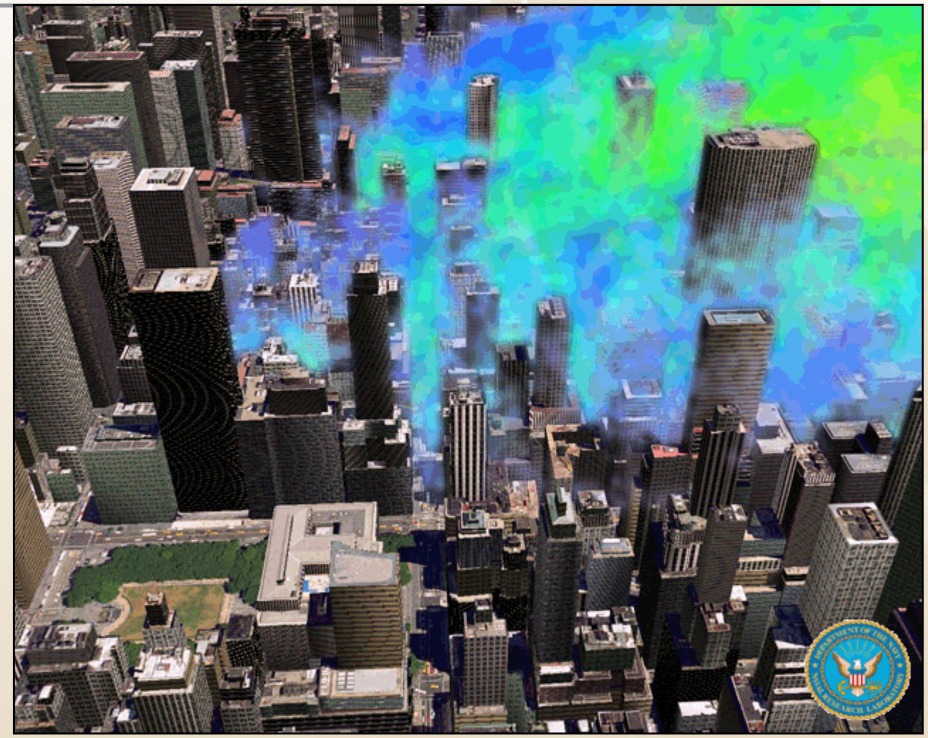
The numerics as modeling approach to turbulence is known as ILES.

- Implicit Large Eddy Simulation relies upon the dissipative properties of modern numerical methods for hyperbolic PDEs to model turbulence.
- At first glance the two fields would seem to have very little in common.
- This approach began through the empirical observation of its effectiveness by pioneers in these modern methods (Boris & Woodward)
- More recently the reasons for the effectiveness of this method are becoming clear through detailed analysis of the dissipative properties of the methods.
- In a nutshell, the methods produce the same dissipative behavior in the inertial range as turbulence.

Starting with Jay Boris, the Naval Research Lab has led this area.



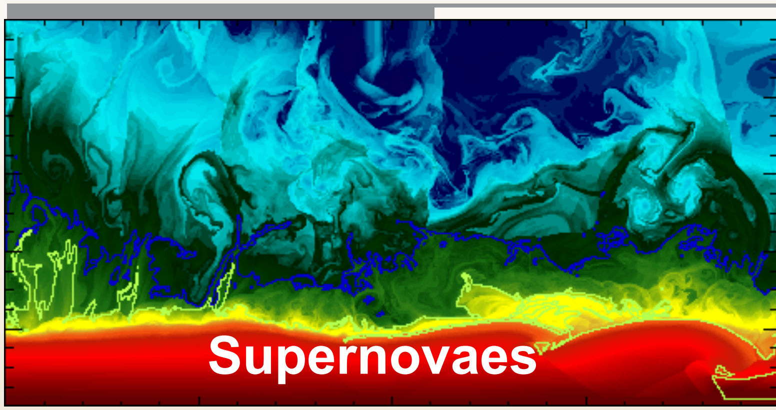
Calculation by F. Grinstein

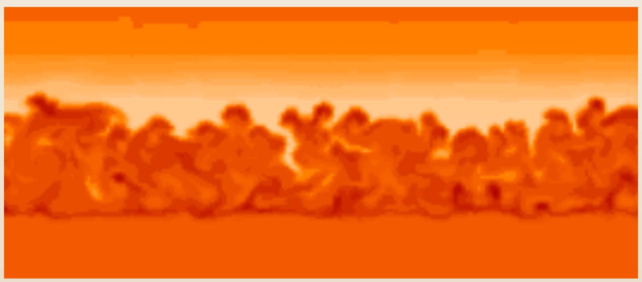


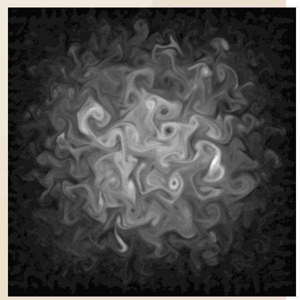
Calculation by Jay Boris and others

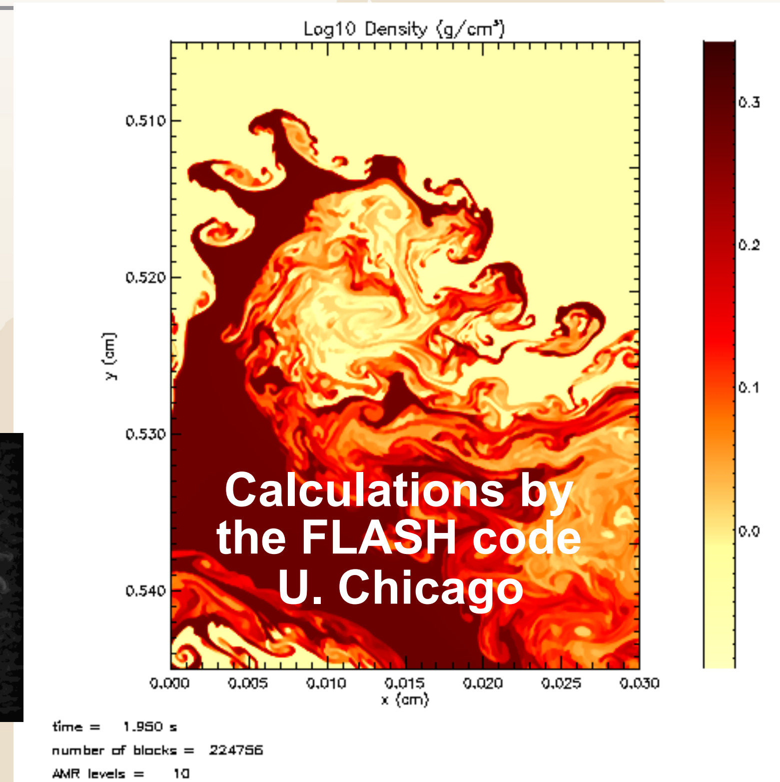
- These calculations use the flux-corrected transport method
  - Many different applications including chemically reactive flows

The PPM method has become the standard in astrophysics and also successful with ILES.



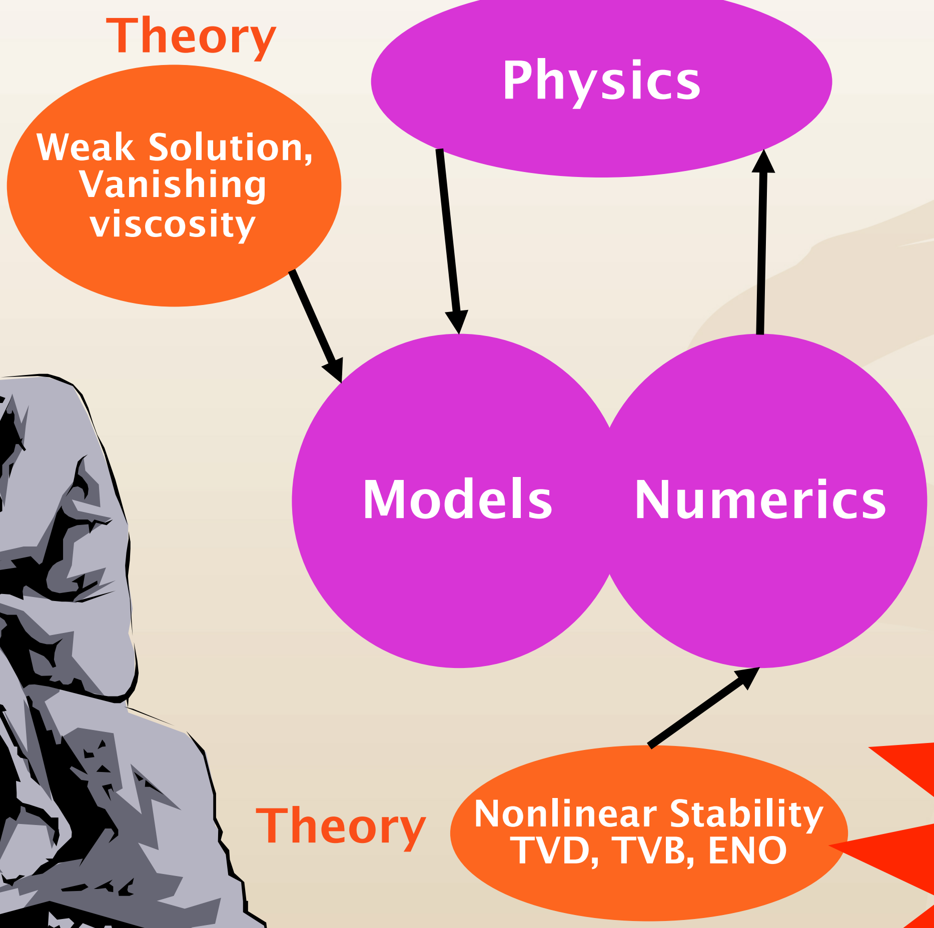






- Turbulence is ubiquitious in astrophysical flows: supernova, natural convection in stars, galactic clouds, cosmology
- PPM is a high-order Godunov method.

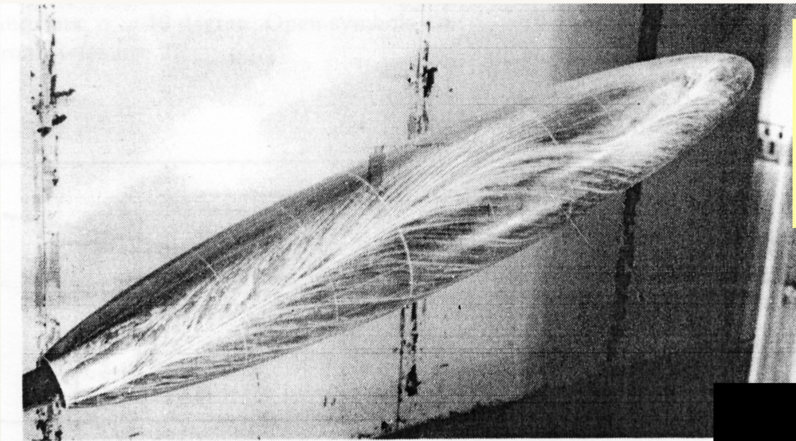
The numerics as modeling modeling is based on different assumptions. Are they less valid than the classical assumptions?



These methods are broadly known as non-oscillatory methods and use non-trivial nonlinear differencing.

Assures that the effective model is stable

More and more engineering flows have been addressed by ILES methods

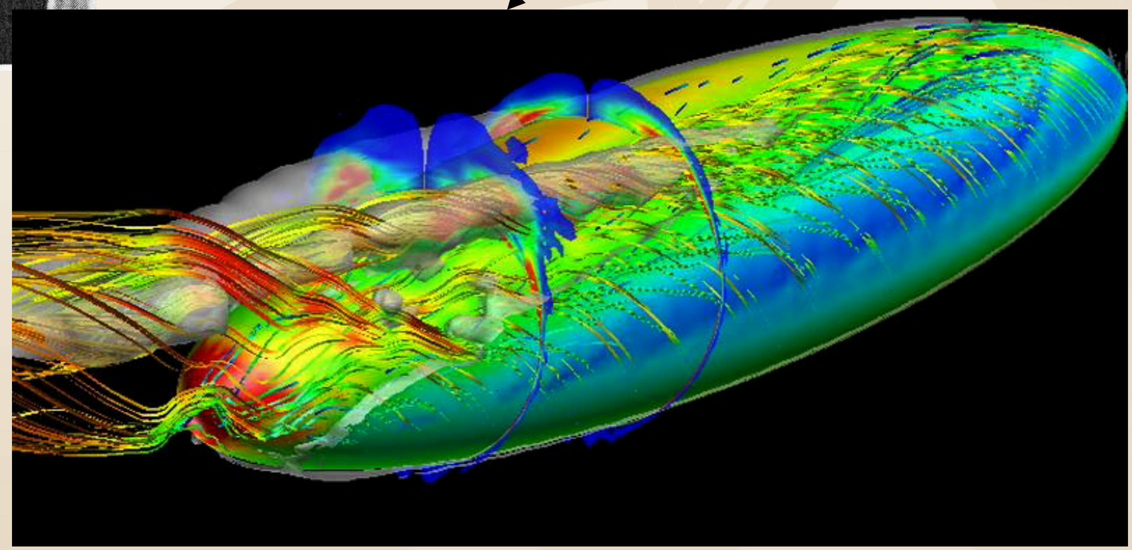


Experiments by:
• Simpson et al (VATech., 1990 -)
• Kreplin & Meier (DLR)
• Hahn & Patel (1979)

$Re_L=4.1\cdot10^6$, a=20, Grid of $1.15\cdot10^6$ cells, $y^+\approx20$, MILES+WM
Fureby et al., AIAA J. ’04

• L=1.36 m, D=0.23 m
• U_0=46 m/s
• Re=$4\cdot10^6$
• angle of attack a=0°–>30°

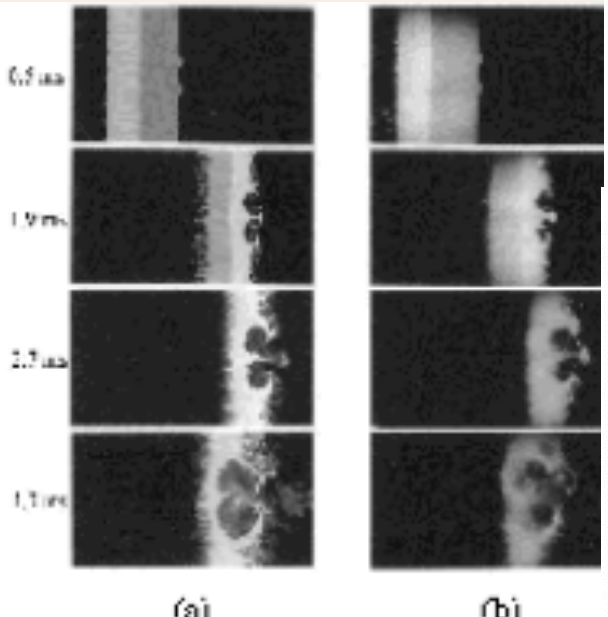
typical MILES studies:
grids 0.8 - $1.6\cdot10^6$ nodes
$y^+\approx5$ – 25



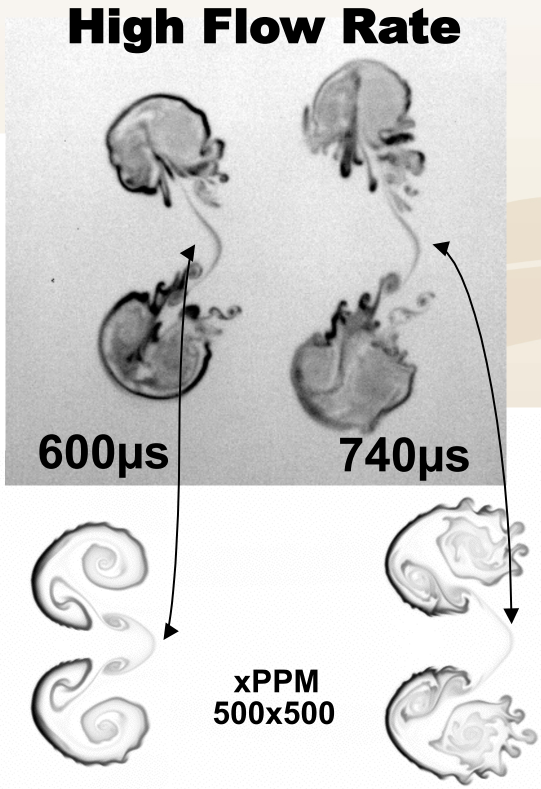
- Fureby has led the validation of these methods for a broad range of engineering flows. Uses TVD schemes

Other applications such as fluid instabilities (R-M & R-T) and atmospheric boundary layers

D.L. Youngs, AWE, UK
Lagrangian (vNR),
Eulerian, 3rd order VanLeer



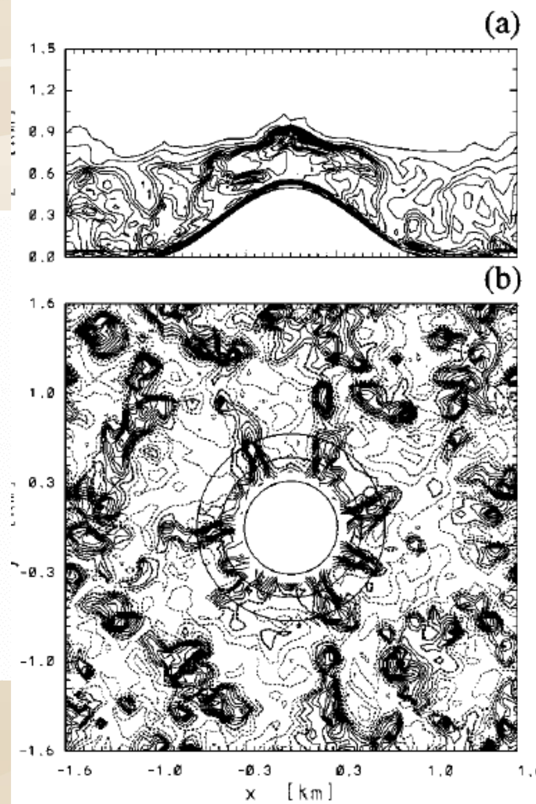
Simulation Experiment



Calculation by W. Rider

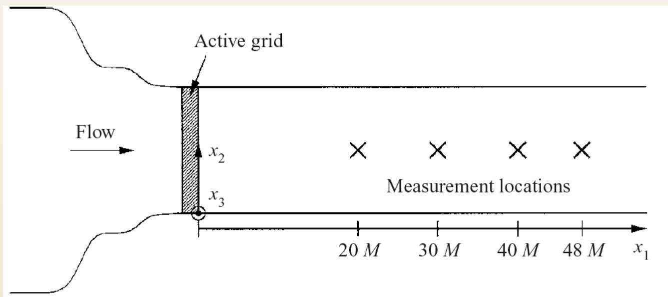
Experiments Tomkins, Prestridge, …LANL

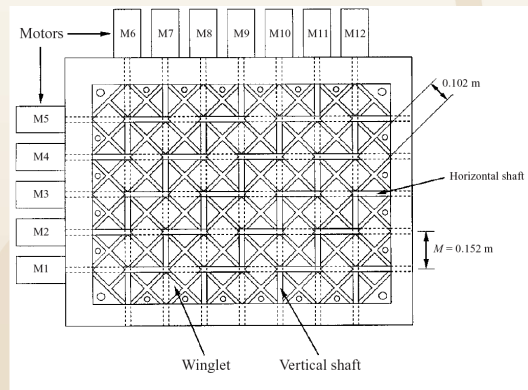
Sorbjan, Smolarkiewicz And Margolin, MPDATA

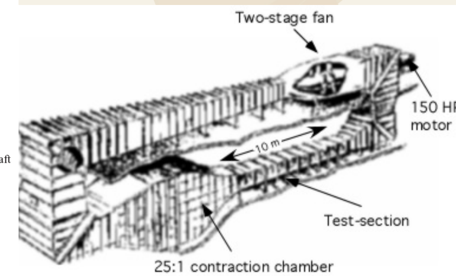


The John Hopkins University experimental setup.

- Data is taken at four stations in the wind tunnel.

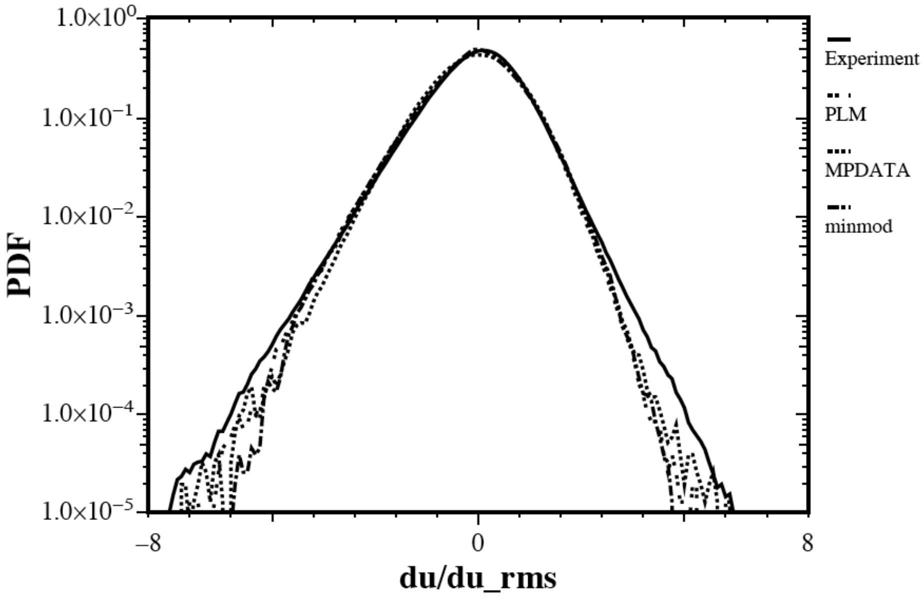


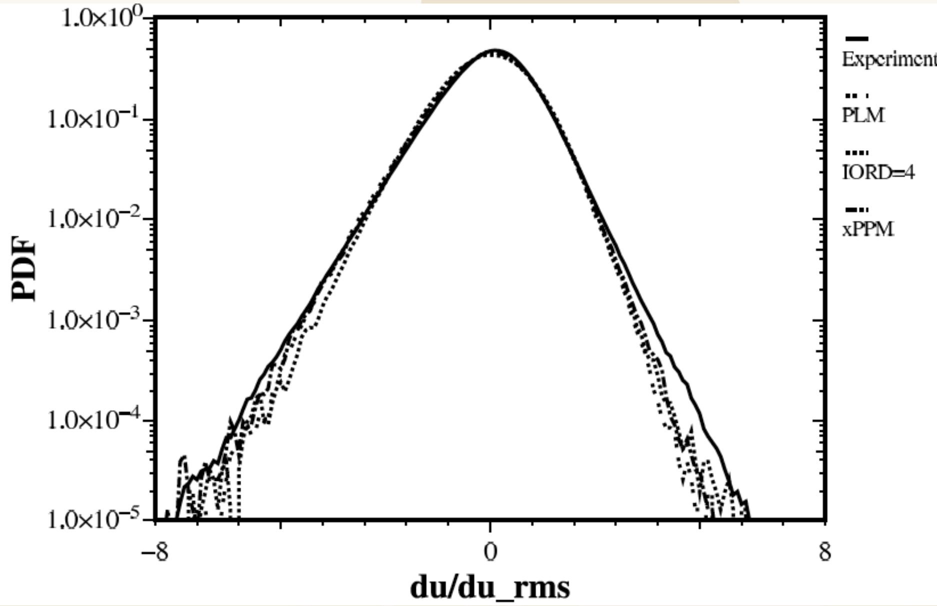




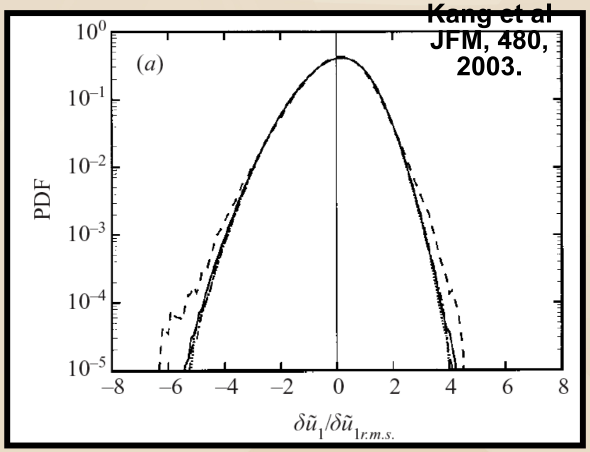
- The initial data used in the LES runs is given by the experimental group and JHU.
- The experimental data was published in JFM (480, pp. 129-160, 2003) and can be found on a JHU website.
- Updates and extends the CBC experiment/data

Results: longitudinal PDF of velocity increments





All the ILES methods produce much more intermitent results than the CLES. The xPPM and MPDATA results are the closest to the data.

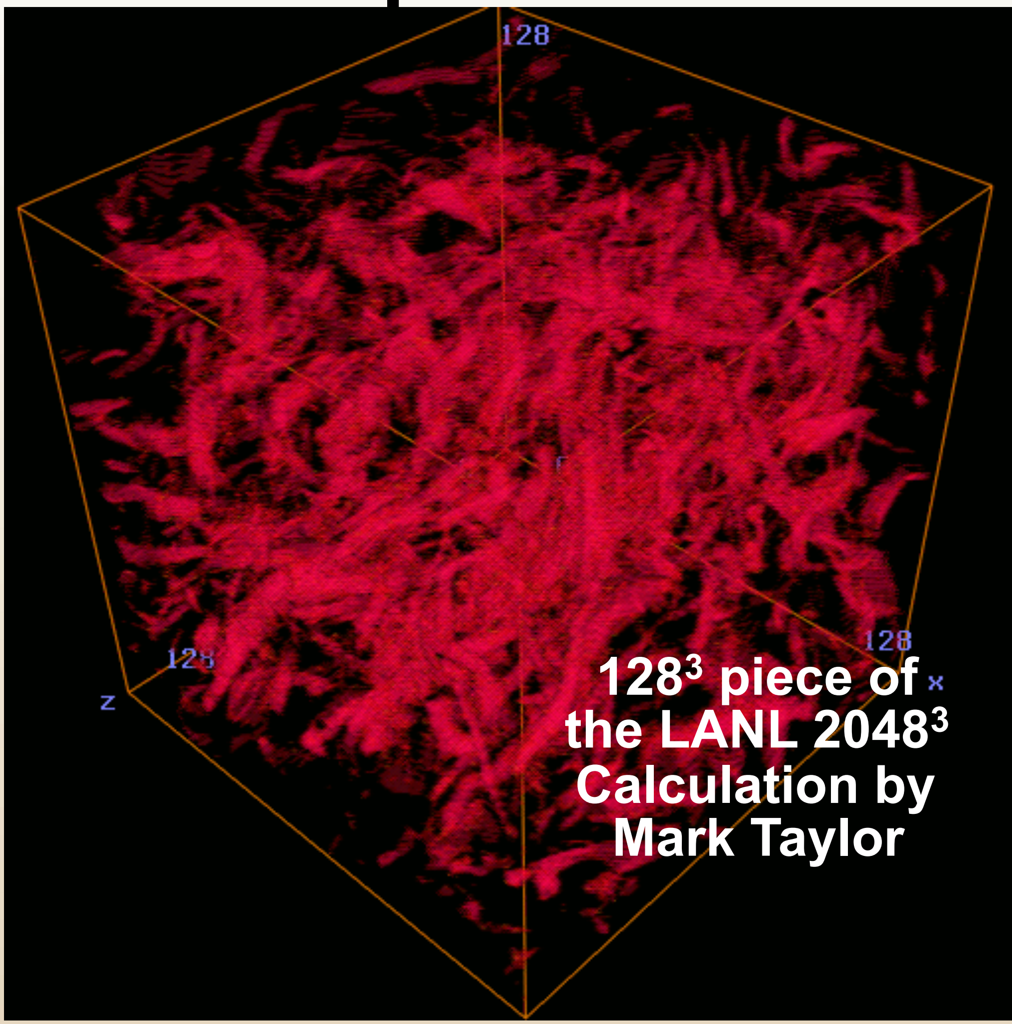


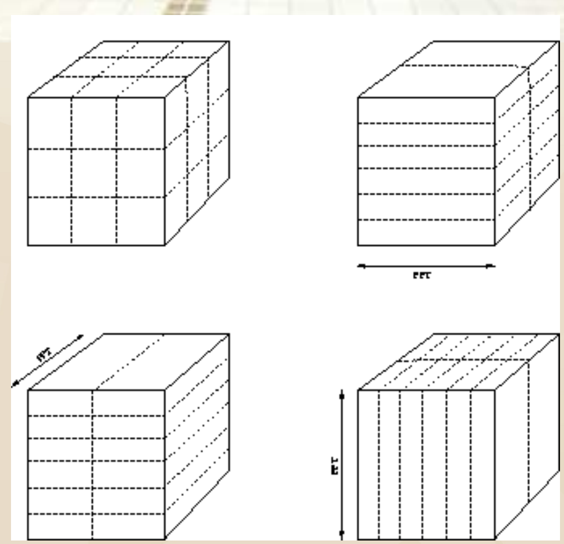
Kang et al JFM, 480, 2003.

Finally there is direct numerical simulation (DNS) of turbulence.

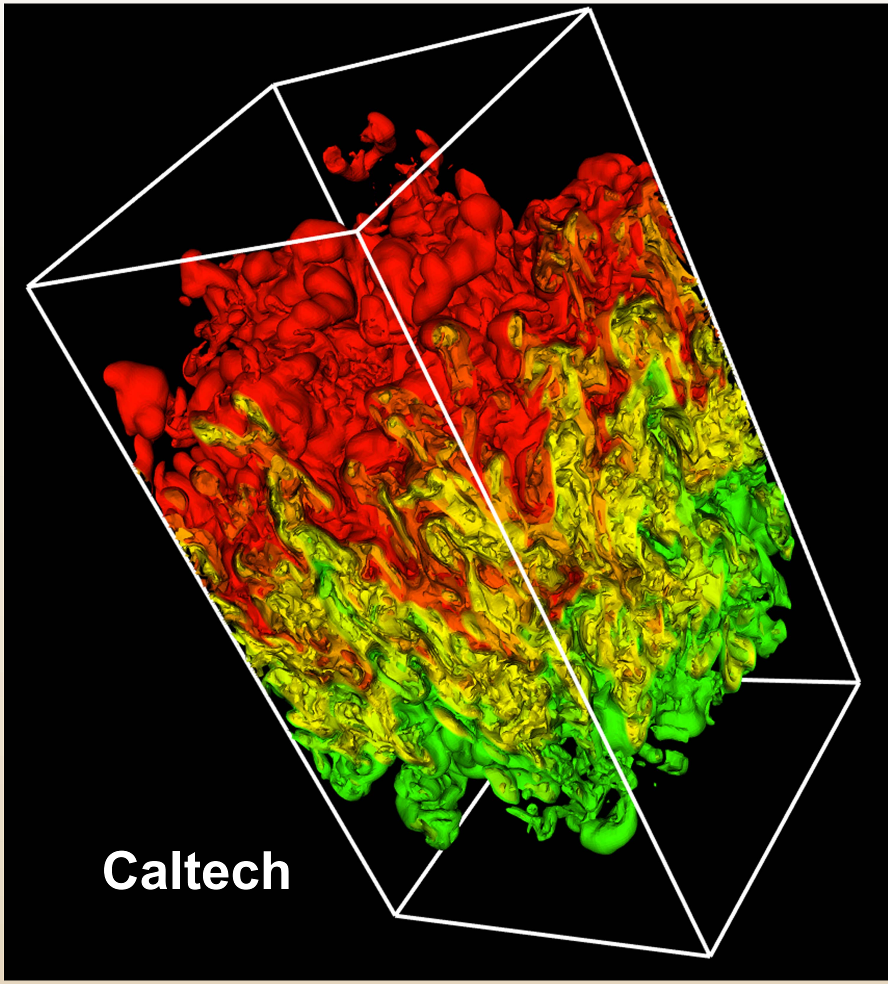
- In DNS all the (relevant) length scales of the flow are “resolved” numerically, i.e. the intent is that the numerical errors are much smaller than any physical effect.
  - These calculation generally receive little verification, but some validation.
  - A recent Thesis at Georgia Tech does verification, most DNS calculation are not resolved!
- Typically solves the incompressible Navier-Stokes equations using spectral or high-order finite difference methods.
- The cost of these calculations scales radically with Reynolds number $Re^{9/4}$
- The largest calculation is 4096^3 gives a Reynolds number of about 100,000 (done on the “Earth Simulator” in Japan).

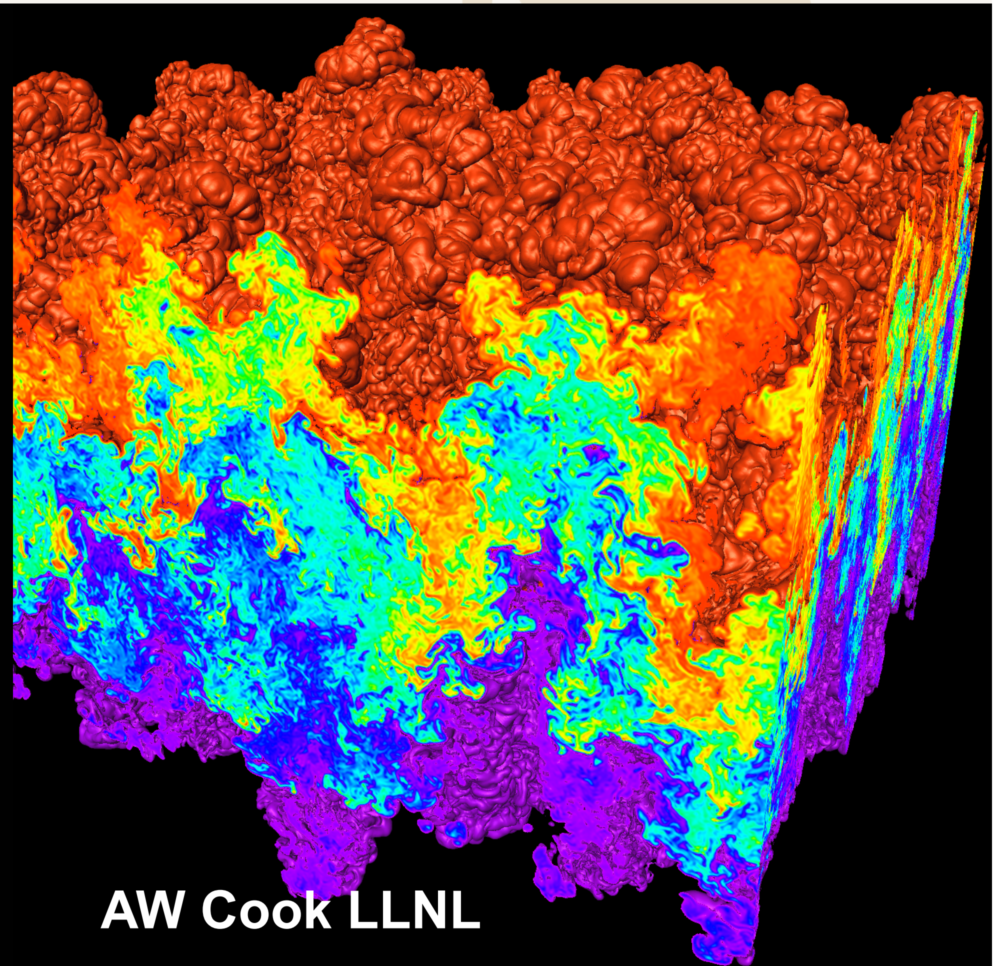
DNS requires extensive data analysis.





Computational studies of R-T are some of the largest simulations ever done.





PART 2. CFD V&V

Quote du jour…

- *"A computer lets you make more mistakes faster than any invention in human history— with the possible exceptions of handguns and tequila."- Mitch Ratliffe*

"Aristotle maintained that women have fewer teeth than men; although he was twice married, it never occurred to him to verify this statement by examining his wives' mouths." - Bertrand Russell

Verification and validation are essential to the quality of simulation.

complementary

- Verification ≈ Solving the equations correctly
  - Mathematics/Computer Science issue
  - Applies to both codes and calculations
- Validation ≈ Solving the correct equations
  - Physics/Engineering (i.e., modeling) issue
  - Applies to both codes and calculations
- Calibration ≈ Adjusting ("tuning") parameters
  - Parameters chosen for a specific class of problems
- Benchmarking ≈ Comparing with other codes
  - "There is no democracy in physics."*

*L.Alvarez, in D. Greenberg, *The Politics of Pure Science*, U. Chicago Press, 1967.

Hieronymus Bosch. 1485

The 7 Deadly Sins of V&V

Otto Dix, 1933

- Assume the code is correct
- Only do a qualitative comparison (e.g., the viewgraph norm!)
- Use problem specific special methods or settings
- Use code-to-code comparisons
- Use only one mesh
- Only show the results that make the code look good - the ones that appear correct
- Don't differentiate between accuracy and robustness

- Lust
- Gluttony
- Envy
- Wrath
- Sloth
- Pride
- Avarice

Traditional "7 Deadly Sins"

7 Virtuous Practices in V&V

- **Assume the code has flaws, bugs, and errors then FIND THEM!**
- **Be quantitative**
- **Verify and Validate the same thing**
- **Use analytic solutions & experimental data**
- **Use systematic mesh refinement**
- **Show all results - reveal the shortcomings**
- **Assess accuracy and robustness separately**

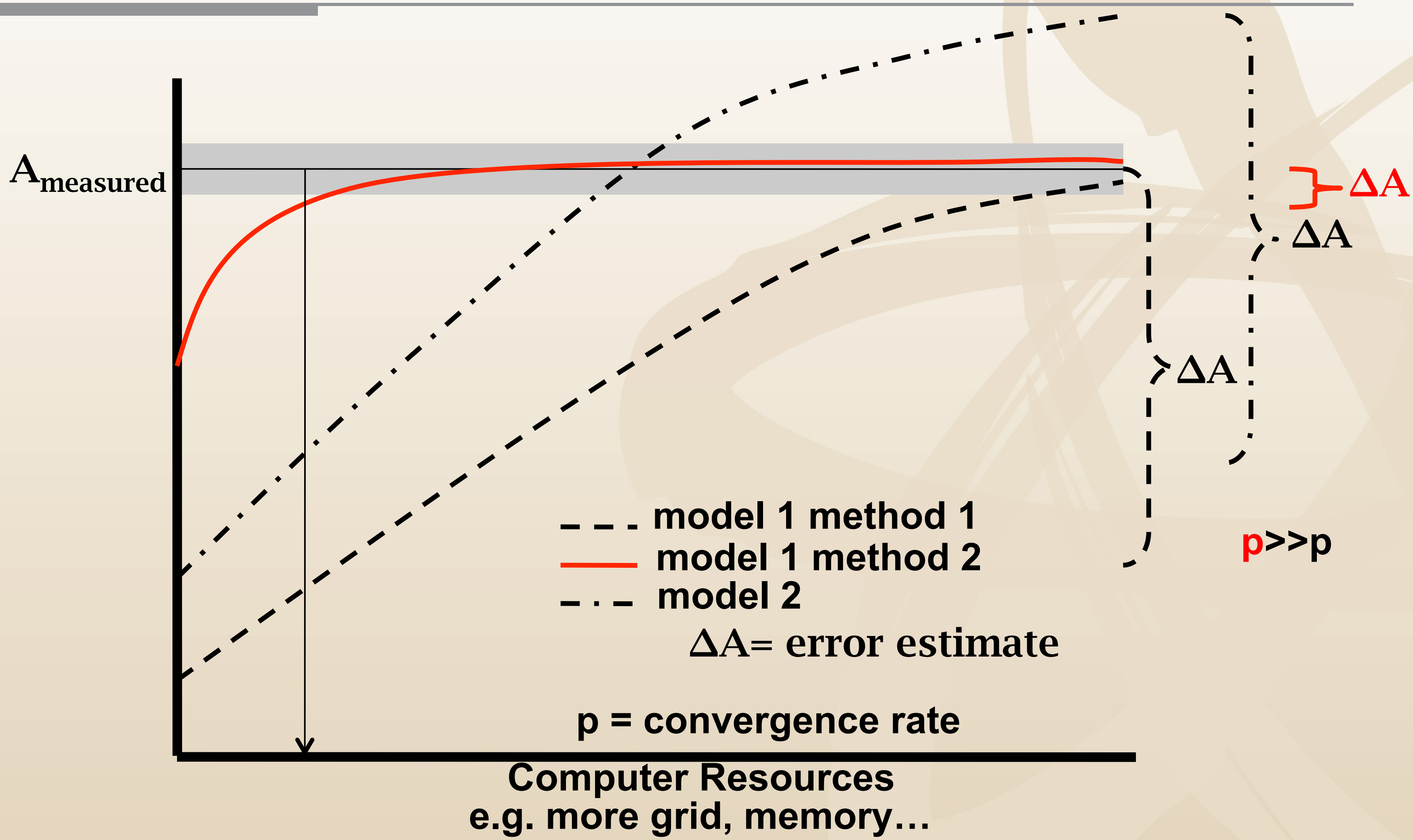
- Prudence
- Temperance
- Faith
- Hope
- Fortitude
- Justice
- Charity

Traditional "7 Cardinal Virtues"

The Corollaries to the Virtues

- V&V helps to ensure quality. We help determine where the codes need to be improved. We help determine the codes' limits. This should help allocate resources.
- Make an unambiguous and clear statement of results. V&V is rigorous and systematic and self-consistent.
- Base results on unambiguous, high quality standards.
- We want codes that are consistent, stable, and convergent. Better computers yield better solutions!
- Show everything, be honest and open.
- Make sure you know what you are looking at.

This schematic shows the sort of information that calculation verification can provide.



VERIFICATION

Verification involves error estimates and computing convergence rates.

- To conduct a verification exercise one needs to compute or rigorously estimate errors.
- These errors are used to compute the convergence rates.
  - The expected rates of convergence depend on the problem solved (how smooth or regular the solution is).
- For a method to be consistent the convergence rate needs to be positive and *in line with expectations for the methods used and the problem solved*.

Types of verification

- **Software**: there is formal verification ala software engineering.
  - I won't have much to say about this
  - Regression testing is a part of this area
- **Code**: means comparing the results of the code with an analytical solution
  - Refine meshes/grids, compute normed errors and convergence rate
- **Solution**: means computing a solution on multiple grids, estimating errors in quantities of interest and the rate of convergence. It is similar to, but not identical to mesh sensitivity.

Code comparisons are frequently suggested as the way to decide how to proceed.

- **This is also benchmarking, and it is dangerous.**
- There is no truth in code comparison… “no democracy in science”.
- One code is often the standard. This status is the result of other tests (verification and validation). The trust is bound up in those problems, not the code.
- The proper way to approach this is to apply the code that is the object of the study to the other tests (not the standard in the code).
- Sometimes there is no other option! But don’t go here first!

Code to Code Comparisons Are a Poor Substitute for Formal Verification

Code Comparison Principle (CCP)

Code 1 = assessed code   Code 2 = benchmark code

What if this term is not negligible?

- Could be that Code 1 models are different from Code 2 models
- Could be a bug in Code 1 or Code 2
- Could be an algorithm flaw in Code 1 or Code 2
- Could be that Code 1 or Code 2 model is not converged

Points to path for better code-to-code comparisons; but if Code 2 is formally verified, why not verify Code 1 to the same verification test suite? And if not, why bother with the code-to-code comparison?

Slide from Marty Pilch's PCMM overview talk

Most verification is built upon this simple error ansatz.

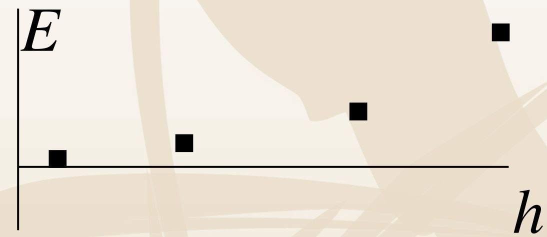
- Here is the simplest way to characterize the error,

$$\|E\|_k = \|S - A\|_k = Ch^{\alpha}$$

  - E is an error measure (norm), S is the numerical solution, A is the “answer”, h is the mesh spacing
- One can get the errors in one of two ways:
  - An exact solution (2 numerical solutions needed), A is the exact solution.
  - Assuming the finer grid is more accurate (3 numerical solutions needed), A is the finer grid solution.

There are different basic models of how the errors will behave.

- Monotone: the best case, the norm for simple problems

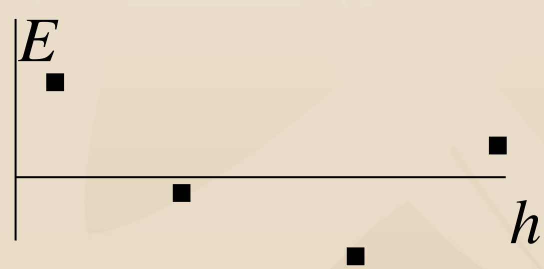


$$E = S - A = Ch^{\alpha}$$

- Bounded: an OK condition, often observed

$$|E| = |S - A| = Ch^{\alpha}$$

- Statistical-Indeterminate: *bad news*, but often observed a problem difficulty increases. Not OK, it's a sign of problems.



$$|E| = \mathbf{PDF}$$

There are several different ways to do a convergence analysis.

- Code Physics Verification: convergence analysis

Hard!

Error in computed = solution

$$\left\| \boxed{f_{\text{exact}}} - f_{\text{comp}} \right\| \sim E_0 + A(\Delta x)^p + B(\Delta t)^q + C(\Delta x)^r (\Delta t)^s + \ldots$$

Convergence rate

Zone size

Spatial and temporal dependence

  - Has demonstrated results with many codes

  - Alternate technology: Method of Manufactured Solutions (MMS)

Continuous $\mathrm{N}\left(f^*\right) = 0$ (Unknown) $\Rightarrow$ Apply $\Rightarrow$ $\mathrm{N}\left(f\right) = g$ (Known) $\Rightarrow$ Project $\Rightarrow$ $\hat{\mathrm{N}}\left(\hat{f}\right) = \hat{g}$ (Computable) Discrete

  - Successfully used for smooth flows
    - Research: MMS for multi-D discontinuous flows

- Calculation Verification

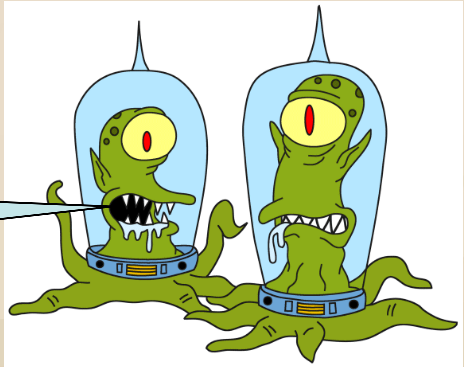
Easy!

$$\left\| \boxed{f_{\text{fine}}} - f_{\text{coarse}} \right\| \sim E_0 + A(\Delta x)^p + B(\Delta t)^q + C(\Delta x)^r (\Delta t)^s + \ldots$$

Error estimates can be computed in many norms and several ways.

- The three most common error norms are the L1, L2 and L infinity norms.
- These are all Lp norms, $$\|E\|_p = \left(\sum_{j=1}^{N} |E|^p \right)^{1/p}$$
- The L1 norm is related to total variation and monotonicity.
- The L2 norm is the energy norm and related to stability in the sense of Hilbert and Banach spaces (eeeiiiiikkkkk!!!!)
- The L infinity norm is really poorly behaved, the largest error in the system.

Through the systematic use of error norms we enslaved the entire galaxy!



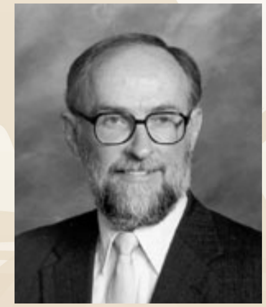
Convergence rates are based on the method and the nature of the problem.

- One can expect to get the full order of accuracy for a method for an ideal test problem where the data begins and remains smooth (continuously differentiable).
- If the problem has a discontinuity or a discontinuous derivative (say a kink), than convergence will be degraded.
- One needs to watch for spontaneously generated discontinuities.

There are expectations.

- For smooth “nice” data a code can be expected to converge at its design order of accuracy.
- With discontinuities you can expect 1st order convergence (at best).
  - Linear discontinuities converge at ($m/(m+1)$), where m is the order of the method.
  - Nonlinear discontinuities converge at 1st order.
- If a problem is grossly under-resolved, non-convergence or even divergence can be observed,
  - As the mesh is refined before normal convergence, the results can show super-convergence, a very high rate.
  - Ultimately the results will settle into the asymptotic result.

The numerical uncertainty can be estimated with various models.



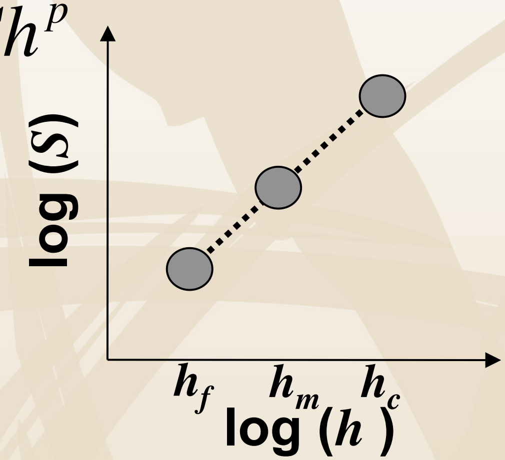
- One model to consider by Roache.
  - This is the Grid Convergence Index (GCI) methodology with a set "safety ratio."

- Another model was proposed by Stern.
  - This model produces a safety factor that depends on both the observed and theoretical convergence rates.
- There are other models, but we believe that these two should be considered primary.
  - Our philosophy is that the focus should be in ***applying*** the estimates to realistic calculations.

Roache's Grid Convergence Index (GCI)* uses a fixed safety factor.

- The standard power error ansatz, $S = A + Ch^p$

$$S = A_k + Ch_k^p;\text{unknowns } S, C, p$$



gives an estimate of numerical error

$$\delta = \frac{\Delta_{mf}}{r_{mf}^p - 1}; \Delta_{mf} = S_f - S_m, r_{mf} = \frac{h_m}{h_f}$$

- A safety factor gives the uncertainty estimate:

$$U_{num} = F_s \delta; F_s = 1.25$$

- This safety factor (supposedly) gives a 95% confidence interval (the consequence of CFD "experience"). Does it apply more generally?

***P. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa(1996).**

Stern's Uncertainty Estimate has a variable "safety factor" or asymptotic correction.

- The estimate developed by Stern uses the same basic framework, but with a key difference…
- The safety factor is not constant, but depends on two pieces of information,
  - The observed order of convergence p_{ob}
  - The theoretical order of convergence p_{th}

$$F_s = \frac{r^{p_{ob}} - 1}{r^{p_{th}} - 1}$$

- This potentially makes it attractive when the computation is not in the asymptotic range,

Testing the estimates against an analytical solution builds confidence.

- The errors can be estimated via calculation verification and exactly using the exact solution.
- This will enable us to examine the quality and safety of the uncertainty estimates.
- We will use three examples:
  - A simple linear ODE
  - A simple linear ODE with “bad” Δt’s
  - Sod’s shock tube

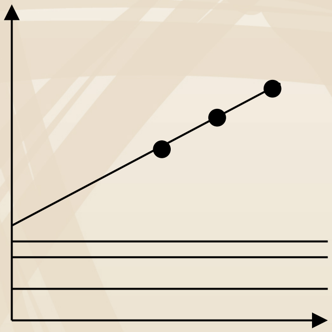
Results for linear ODE

- We'll start with the simplest thing possible,

$$\dot{u} = \lambda u \rightarrow u(t) = u(0)\exp(-\lambda t)$$

  - Use a first-order forward Euler method

$$\begin{array}{ccc} p_T = 1 & \Delta_{cm} = 0.027 & F_{\text{Roache}} = 1.25 \\ p = 1.13 & \Delta_{mf} = 0.012 & F_{\text{Stern}} = 1.18 \\ & \delta = 0.015 & F_{\text{Exact}} = 1.12 \end{array}$$



  - Compare with a second-order modified Euler

$$\begin{array}{ccc} p_T = 2 & \Delta_{cm} = 0.0036 & F_{\text{Roache}} = 1.25 \\ p = 2.16 & \Delta_{mf} = 0.0008 & F_{\text{Stern}} = 1.16 \\ & \delta = 0.0002 & F_{\text{Exact}} = 1.09 \end{array}$$

Results for linear ODE with a bad choice for time step size.

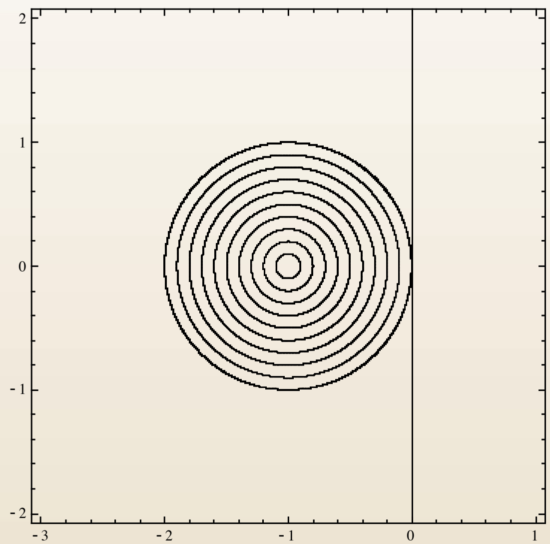
- We'll continue with the simplest thing possible and forward Euler, $\dot{u} = \lambda u \rightarrow u(t) = u(0)\exp(-\lambda t)$
- Use a too large time step, Δt=0.1,

$$p_T = 1 \quad \Delta_{cm} = 0.0022 \quad F_{Roache} = 1.25$$
$$p = 0.44 \quad \Delta_{cm} = 0.0016 \quad F_{Stern} = 2.81$$
$$\lambda = 5 \quad \delta = 0.0045 \quad F_{Exact} = 2.33$$

- Study a "growing" case

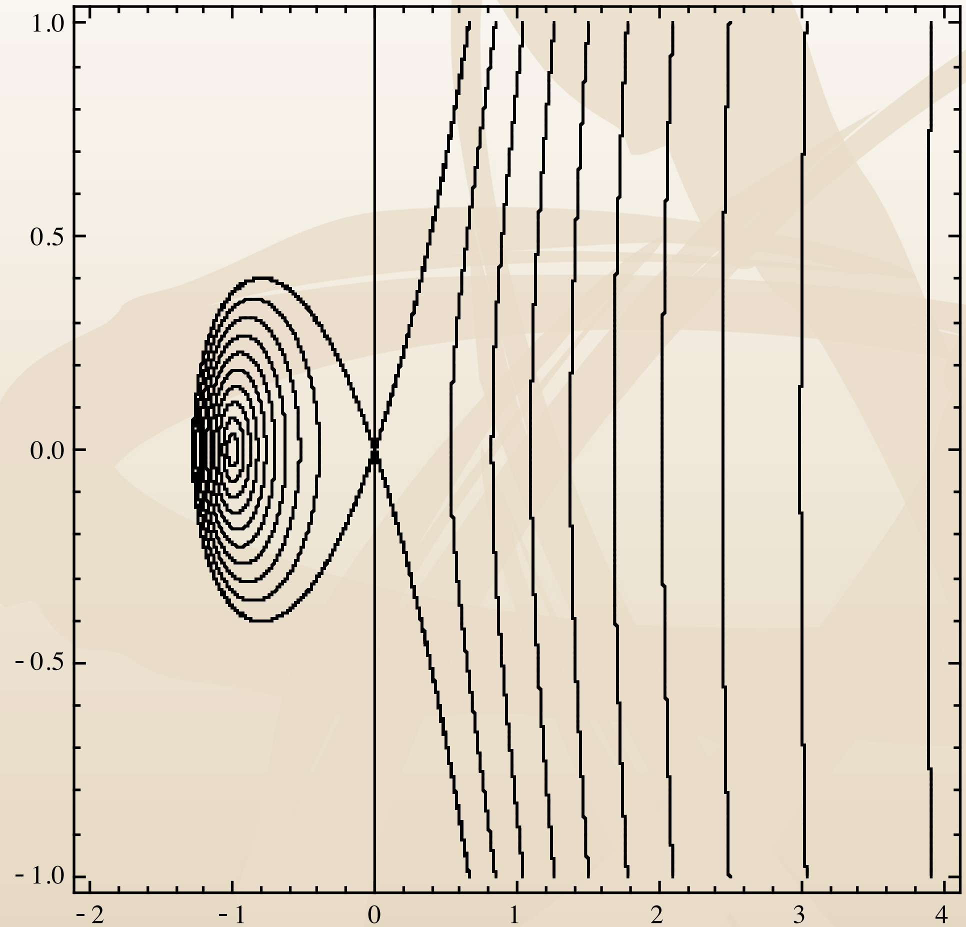
$$p_T = 1 \quad \Delta_{cm} = -29.07 \quad F_{Roache} = 1.25$$
$$p = 0.25 \quad \Delta_{cm} = -24.46 \quad F_{Stern} = 5.31$$
$$\lambda = -5 \quad \delta = -129.9 \quad F_{Exact} = 3.49$$

This can be understood with a bit of numerical analysis



Stability plot

$$\left|1-\lambda\Delta tL\right|\leq 1$$



Order star

$$\frac{\left|1-\lambda\Delta tL\right|}{\exp(-\lambda\Delta t)}\leq 1$$

Results for numerical UQ estimation with Sod's Shock Tube.

- Sod's shock tube uses an ideal gas with a pressure ratio of 10 and a density ratio of 8
  - Solve this with a Godunov-type method

$$p_T = 4/5 \quad \Delta_{cm} = 1.21 \times 10^{-3} \quad F_{\text{Roache}} = 1.25$$
$$p = 1.19 \quad \Delta_{mf} = 5.28 \times 10^{-4} \quad F_{\text{Stern}} = 1.74$$
$$\delta = 4.09 \times 10^{-4} \quad F_{\text{Exact}} = 1.39$$

Density

$$p_T = 1 \quad \Delta_{cm} = 8.93 \times 10^{-4} \quad F_{\text{Roache}} = 1.25$$
$$p = 1.13 \quad \Delta_{mf} = 4.07 \times 10^{-4} \quad F_{\text{Stern}} = 1.19$$
$$\delta = 3.52 \times 10^{-4} \quad F_{\text{Exact}} = 0.82$$

Pressure

Example, Combined Space-Time Convergence Analysis

- Consider the following error Ansatz:

$$\mathrm{F}\left(\xi;\Delta x,\Delta t\right) = \left\| f_{\text{exact}} - f_{\text{comp}} \right\| \sim A\left(\Delta x\right)^{p} + B\left(\Delta t\right)^{q} + C\left(\Delta x\right)^{r}\left(\Delta t\right)^{s}$$

Seven **unknowns** ➡ **Seven** **equations** **required**

$$\mathrm{g}\left(\xi;\Delta x_n,\Delta t_n\right) = \left\| f_{\text{exact}} - f_{\text{comp}} \right\| - \xi_1\left(\Delta x_n\right)^{\xi_2} - \xi_3\left(\Delta t_n\right)^{\xi_4} - \xi_5\left(\Delta x_n\right)^{\xi_6}\left(\Delta t_n\right)^{\xi_7}$$

$$\mathrm{g}\left(\xi;\Delta x_n,\Delta t_n\right) = 0\ ,\quad n=1,\ldots,7 \quad \Rightarrow \quad \boxed{\mathrm{G}\left(\xi\right) = 0}$$

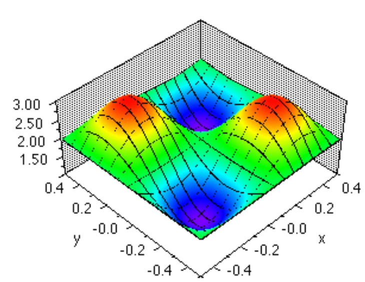
Obtain solutions with generalized Newton's method

  – **Strength: Assumption regarding combined error sources**

  – **Weakness: Complexity, cost, uncertainty in solution**

    • **Example: 2D linear advection**

Analysis of problem involving ***nonlinear*** *fields is in progress…*



	A	p	B	q	C	r	s
Set 1	0.010	1.90	0.0067	1.95	0.010	0.90	0.90
Set 2	0.010	2.00	0.0078	1.97	0.010	1.01	1.00

What happens when codes don't converge.

- Start simplifying the problem:
  - Weaken the jumps or magnitude of problem difficulty,
  - Take the problem to asymptotic limits (strong shock or weak shock limit, etc…)
  - Change the problem in small ways
  - Refine the grid some more (is the grid sufficient?)
- If all else fails admit that there is a problem that can't be fixed without going deeper.

Begin expecting methods to fail, don’t begin expecting them to succeed.

- The best way to proceed with a testing (verification) study is to assume that something is wrong with the code and prove what the problem is.
- If you cannot prove that the code has an error than the code is more likely to be correct.
- The code is only correct to the extent that the testing covers the domain of interest.

Its important to always remember the theoretical expectations.

- The Lax equivalence theorem: consistency & stability equals convergence
- The Lax-Wendroff theorem: conservation is required to assure weak solutions.
- The Hou-LeFloch theorem: without conservation you will not get weak solutions
- The Majda-Osher theorem: first-order accuracy with discontinuities.

VALIDATION

Quote du jour…

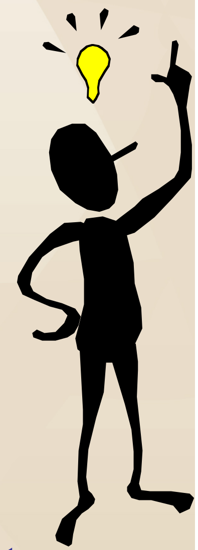
"The purpose of computing is insight, not pictures"–Richard Hamming

"Dilbert isn't a comic strip, it's a documentary" – Paul Dubois

A new proposed definition for Validation

Validation is the process of assessing the quality of modeling a physical process and the magnitude of error associated with the simulation (including numerical error, verification).

*Validation is determining whether you are solving the **correct** model (as well as how well)*

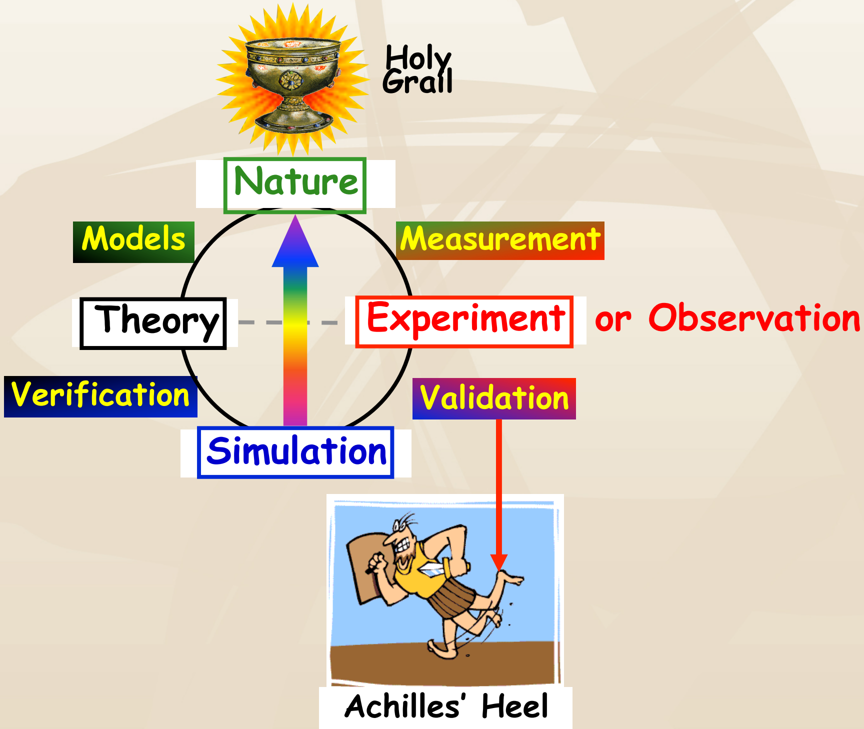


- The benefit of this definition is subtle
  - The appropriateness of a model for a physical circumstance is central.
  - The fact that even an appropriate model has errors (uncertainty) is defined.
  - This process must include model verification as a key part of the complete validation.

The issues with experimental connections are essential to avoid!

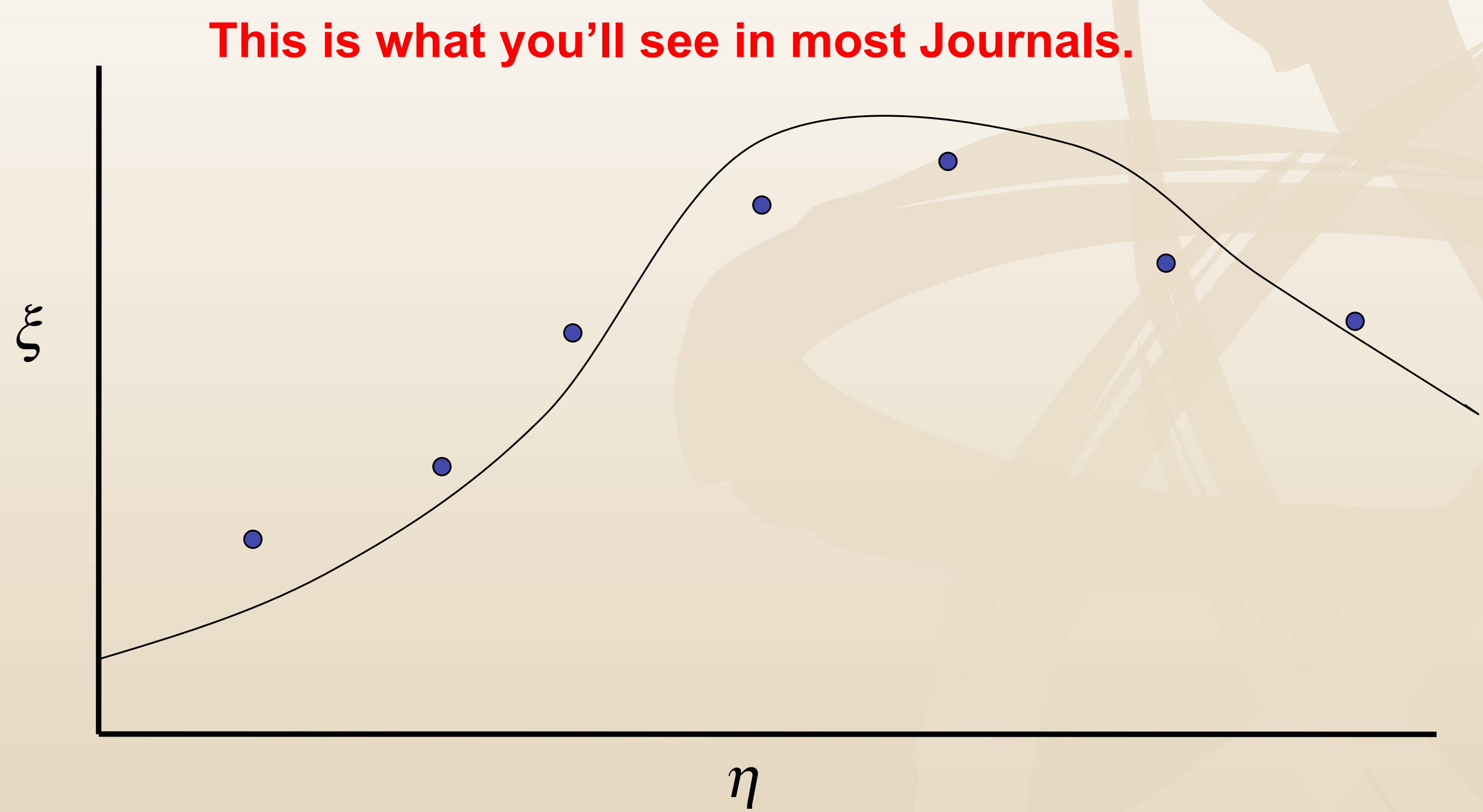
- Validation depends on experiment and measurement.
- The choice to develop separate experimental & computational programs is a mistake, they must be conducted together.
- The assessment of modeling quality needs to consider the quality of the measurement.
  - Bad measurements mean poor constraints for modeling.

A conceptual picture of V&V within the context of science



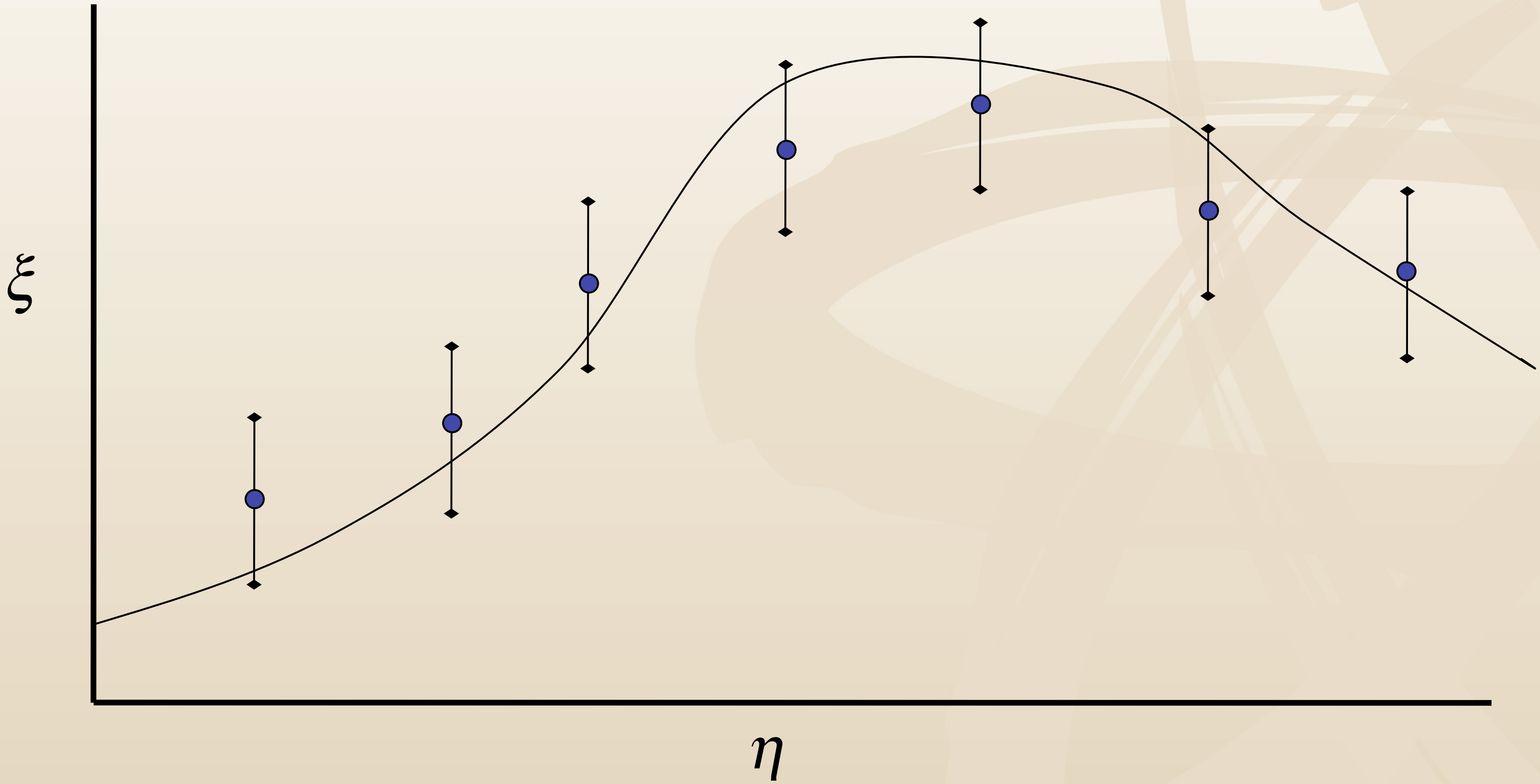
Code = Theory
Simulation = Analysis

This is the way validation is usually presented in the literature.

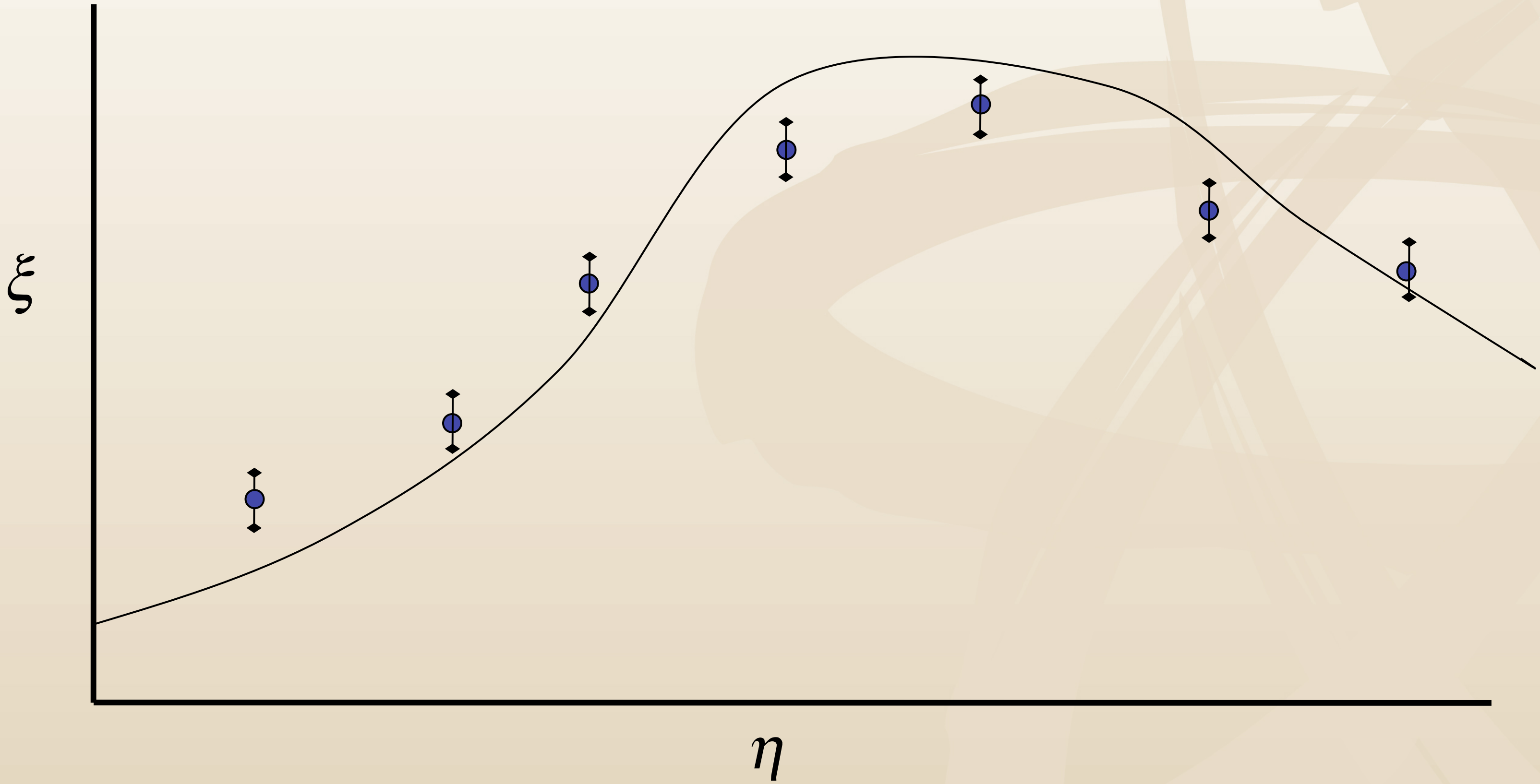


This presentation is an improvement because experimental error is shown.

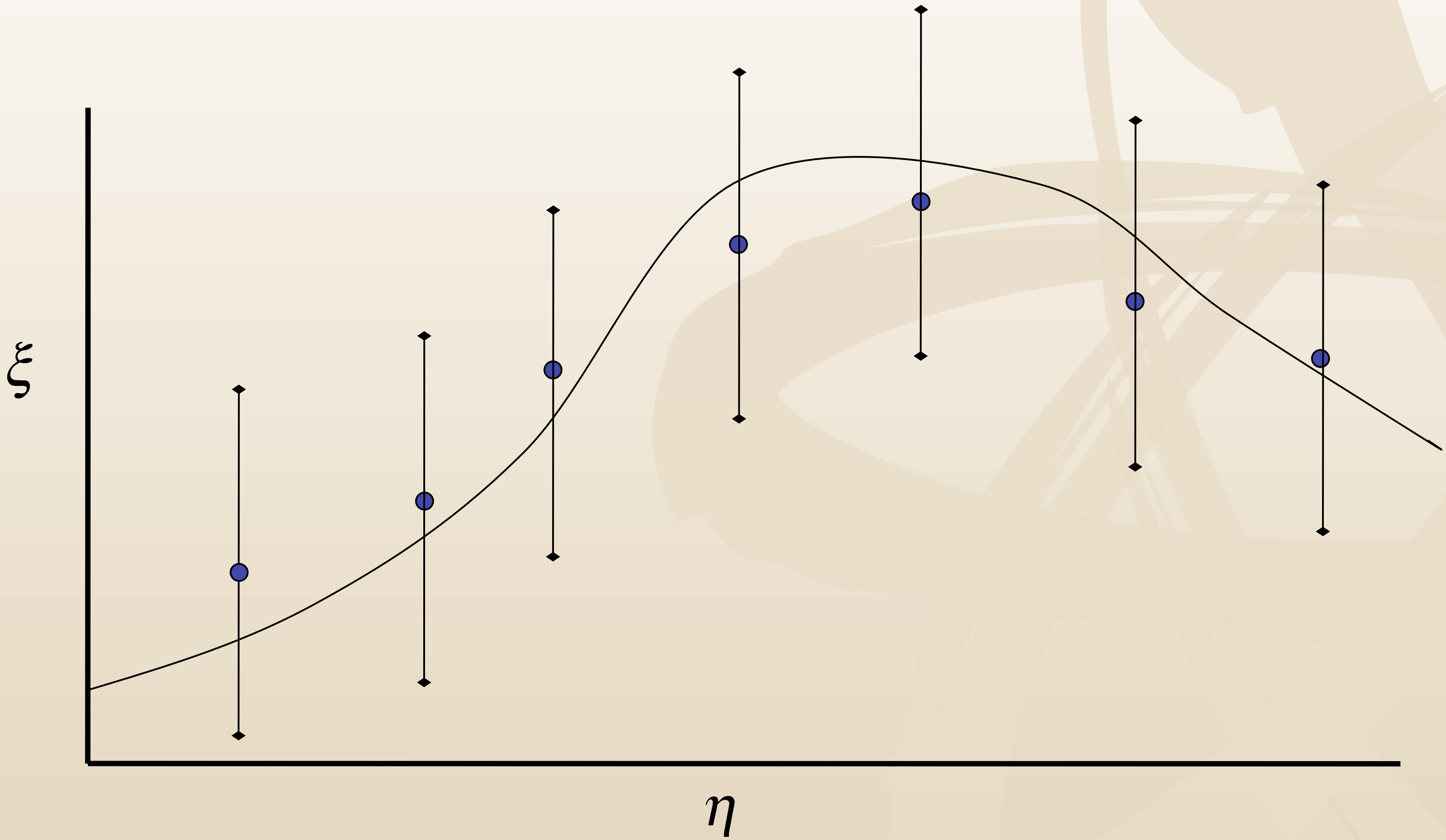
This is *not* what you'll see in most Journals, but you should.



The previous slide showed a "good" agreement with data, this is a "poor" one!

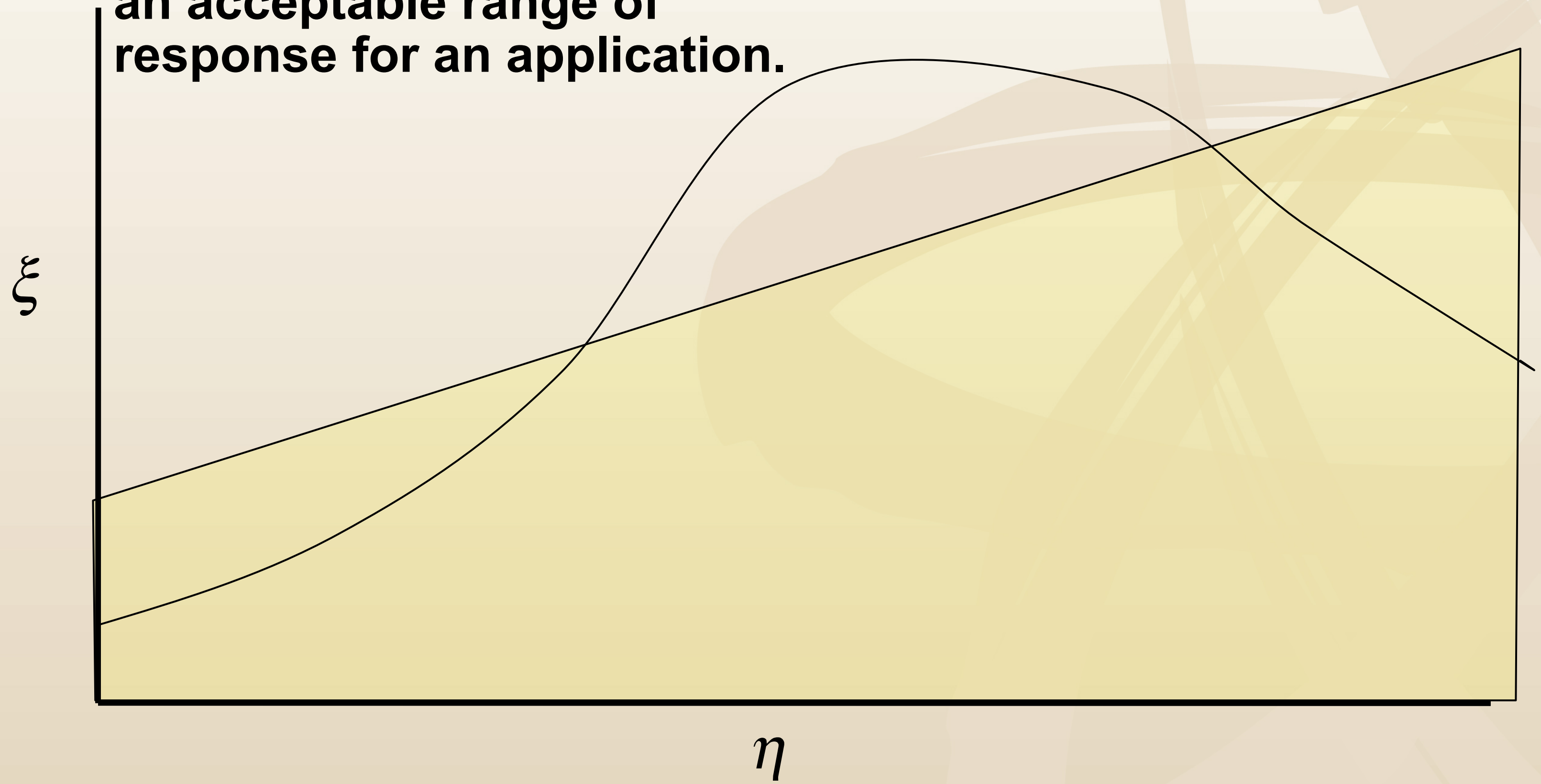


The previous slide showed a "poor" agreement with data, this is a "great" one!



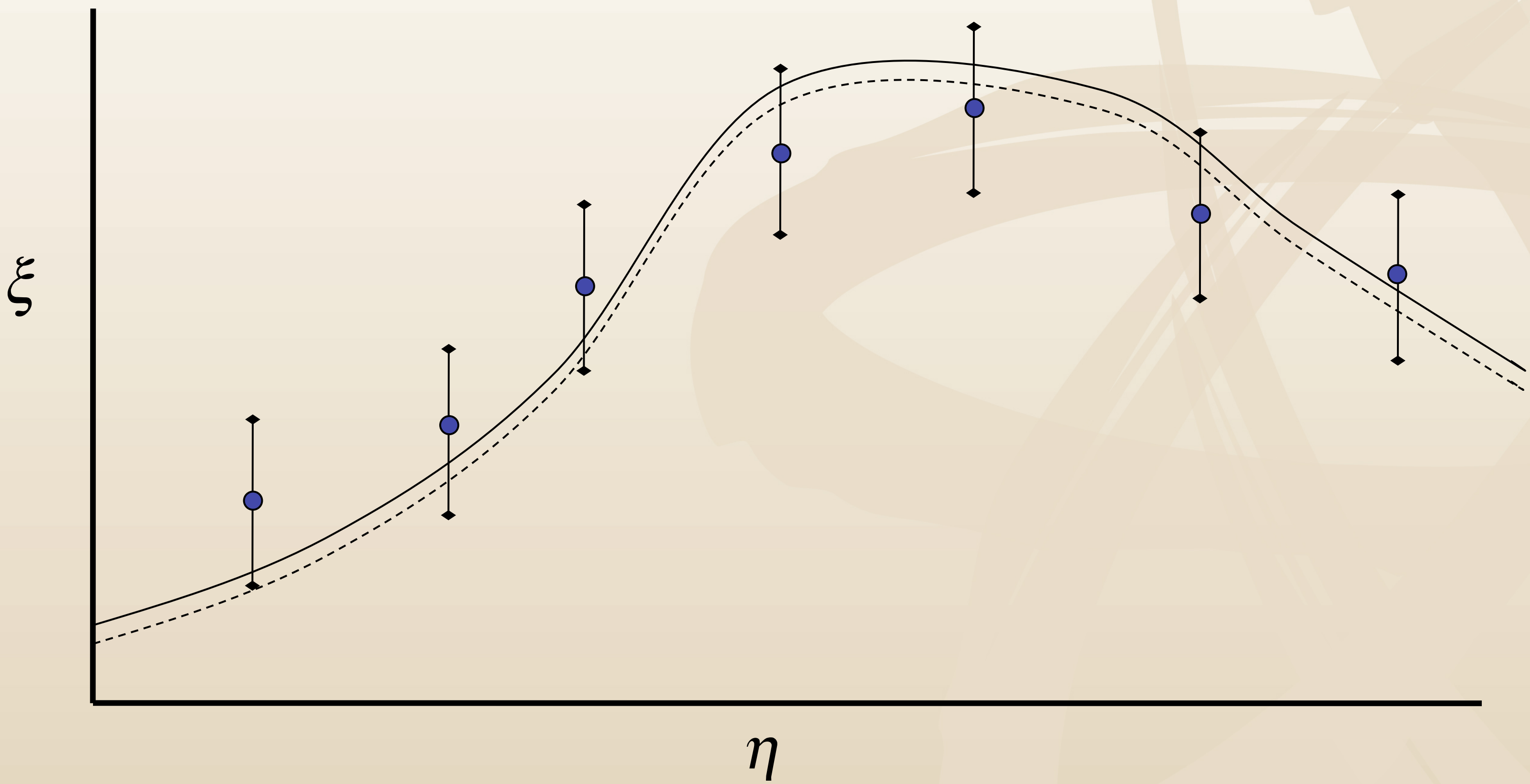
What if the errors in the data are not known? Use another accuracy scale

Here the yellow shape shows an acceptable range of response for an application.



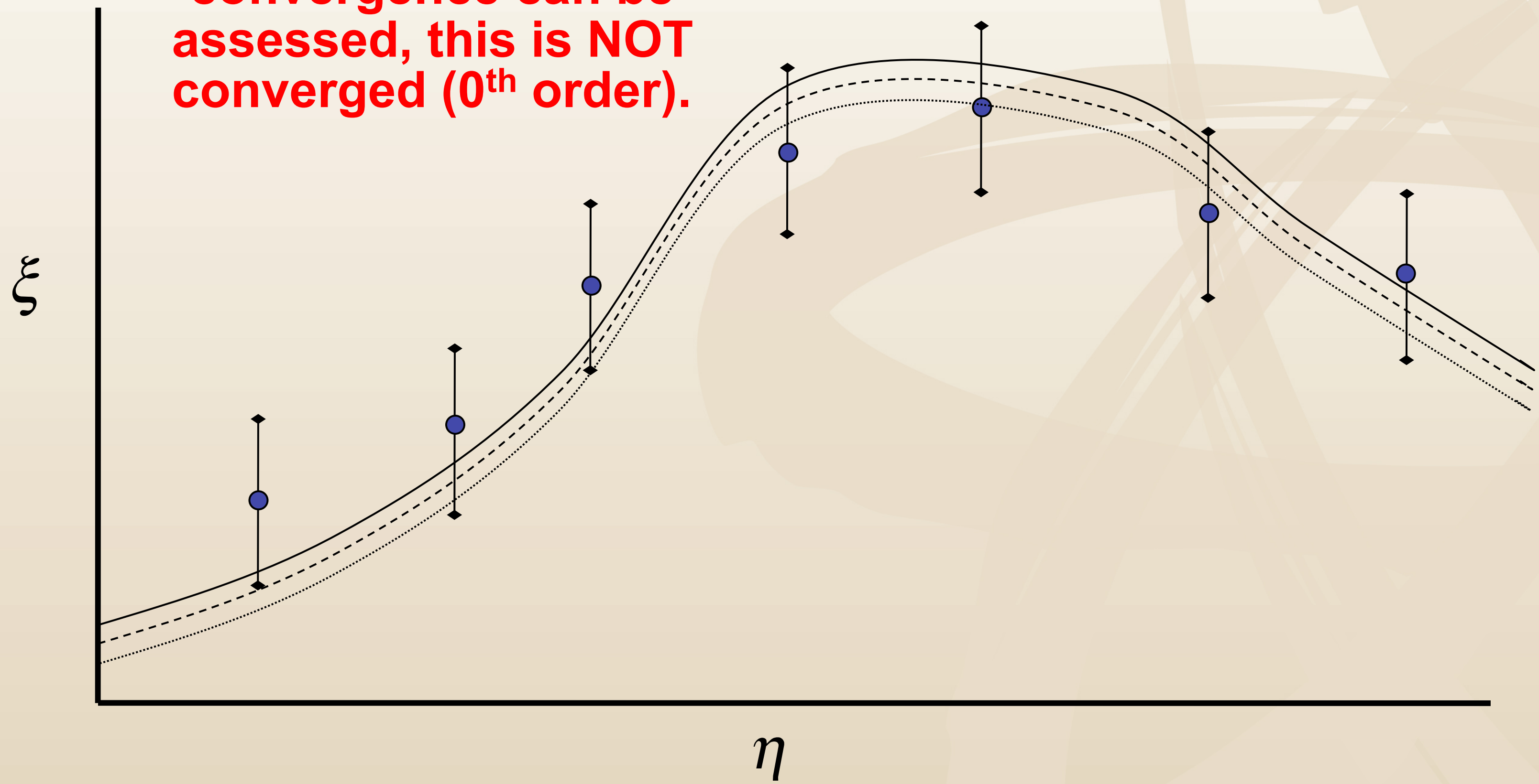
Here is a notion of how a "converged" solution might be described.

You might see this although rarely depicted in this manner.

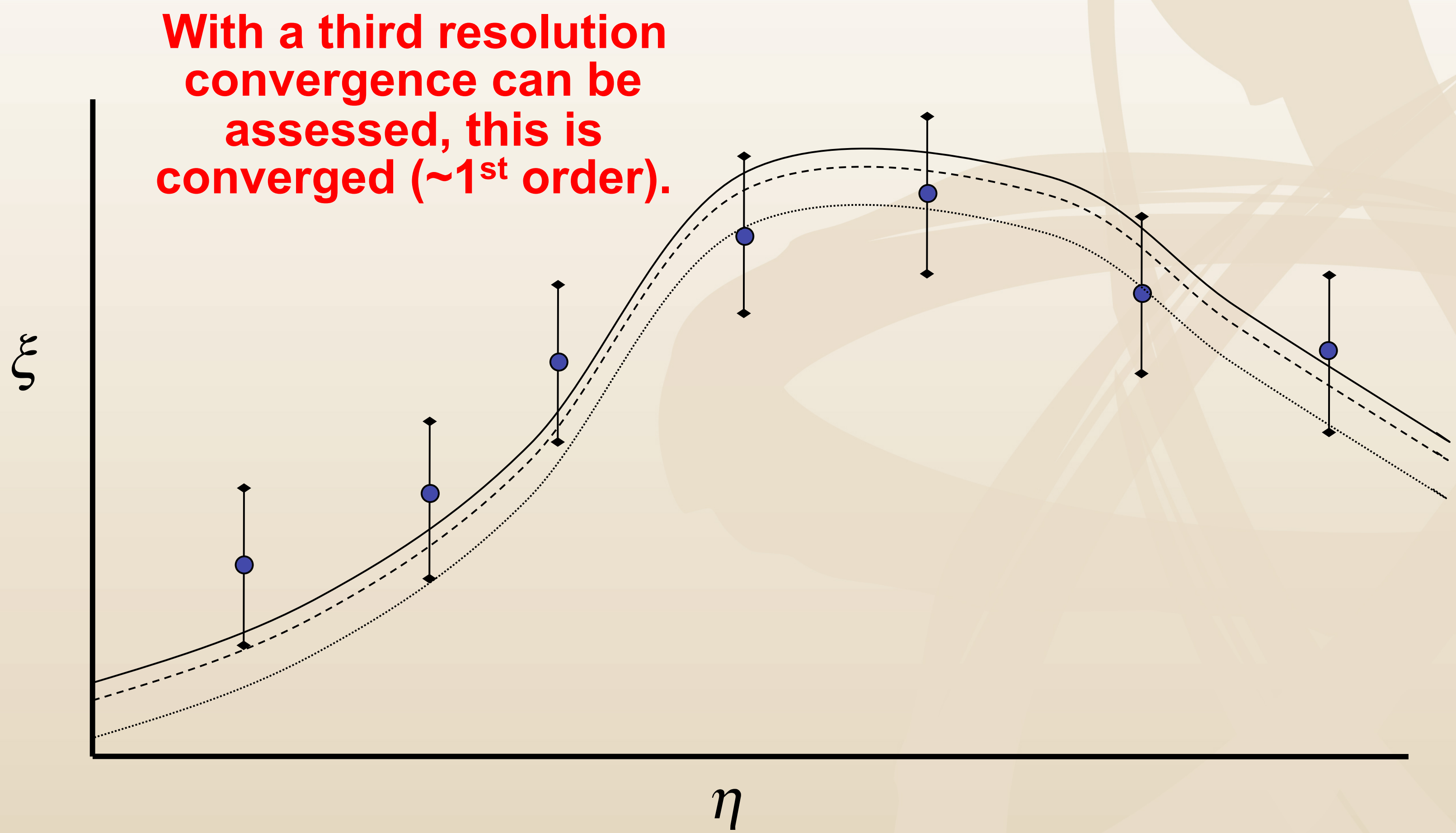


Here is a notion of how a "converged" solution might be described.

With a third resolution convergence can be assessed, this is NOT converged (0th order).

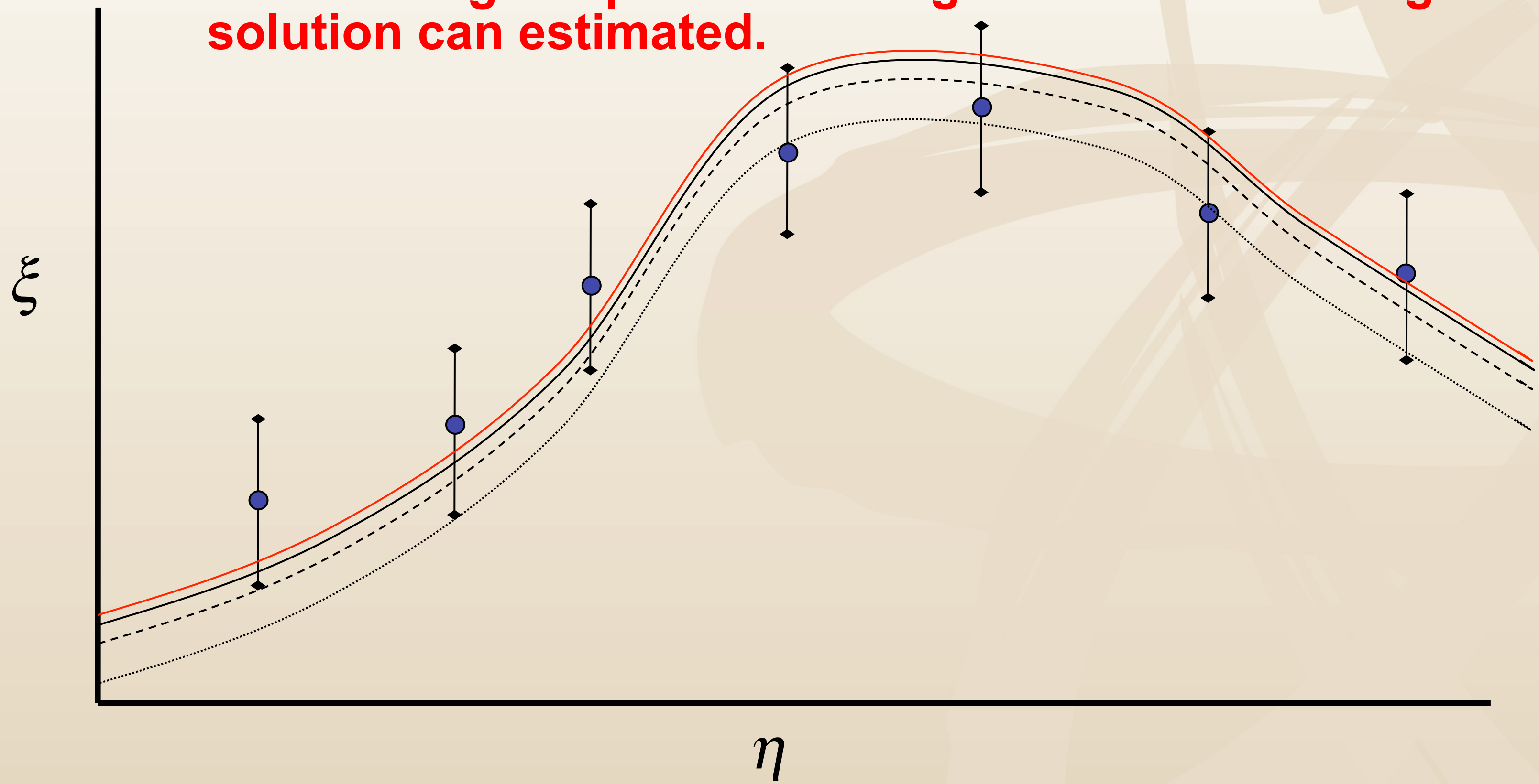


Here is a notion of how a “converged” solution might be described.

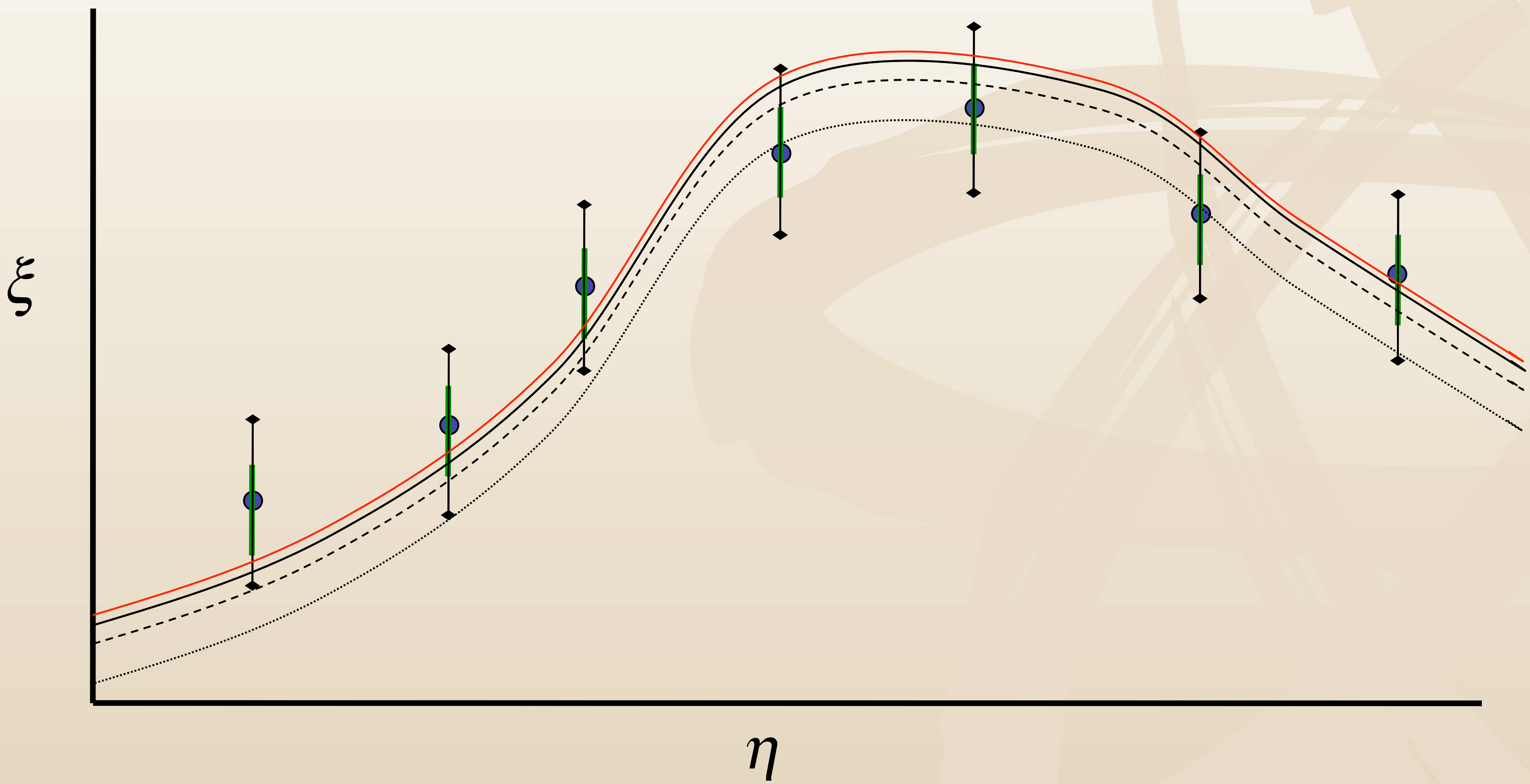


This sequence of meshes can be used to extrapolate the solution.

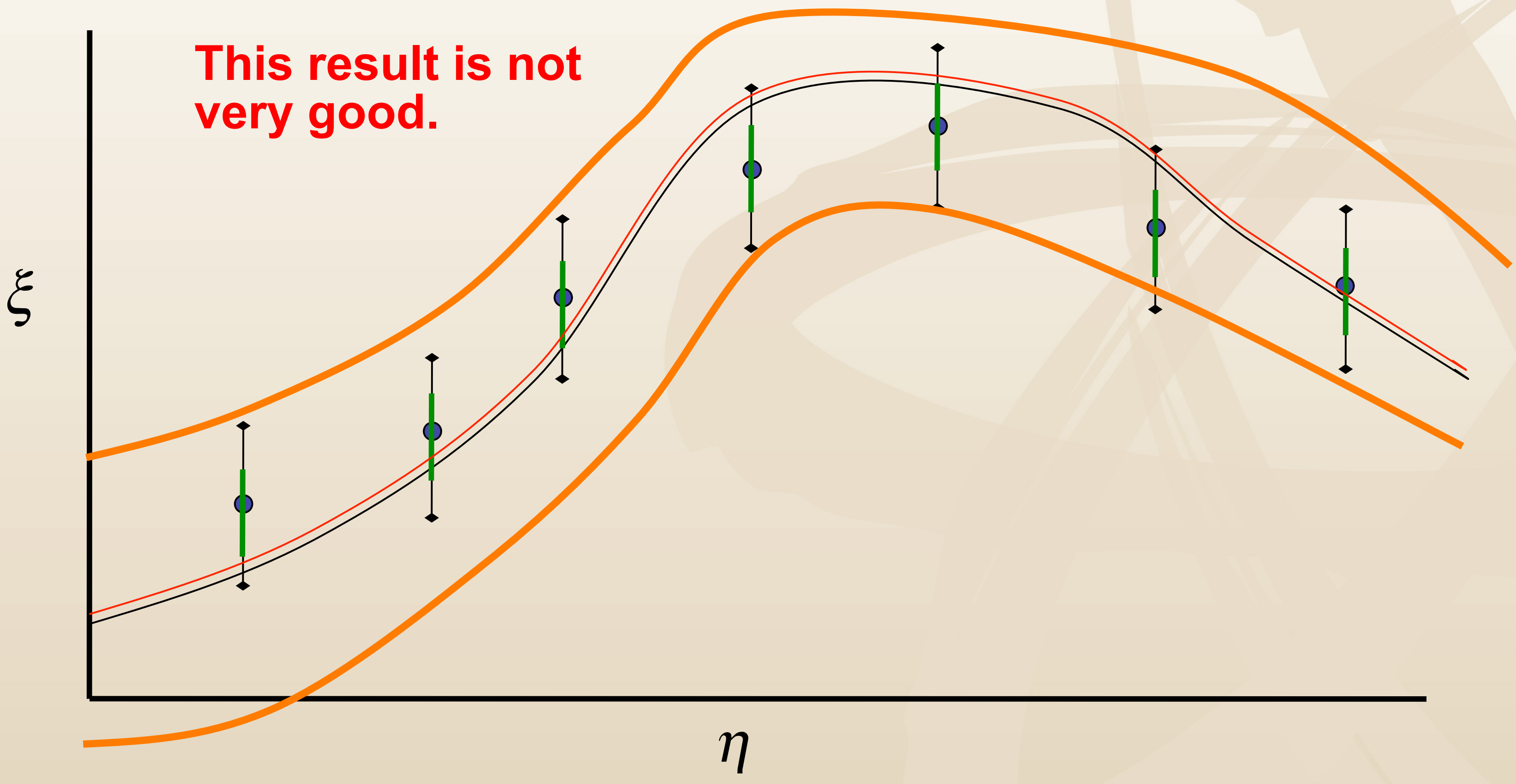
With three grids plus a convergence rate a converged solution can estimated.



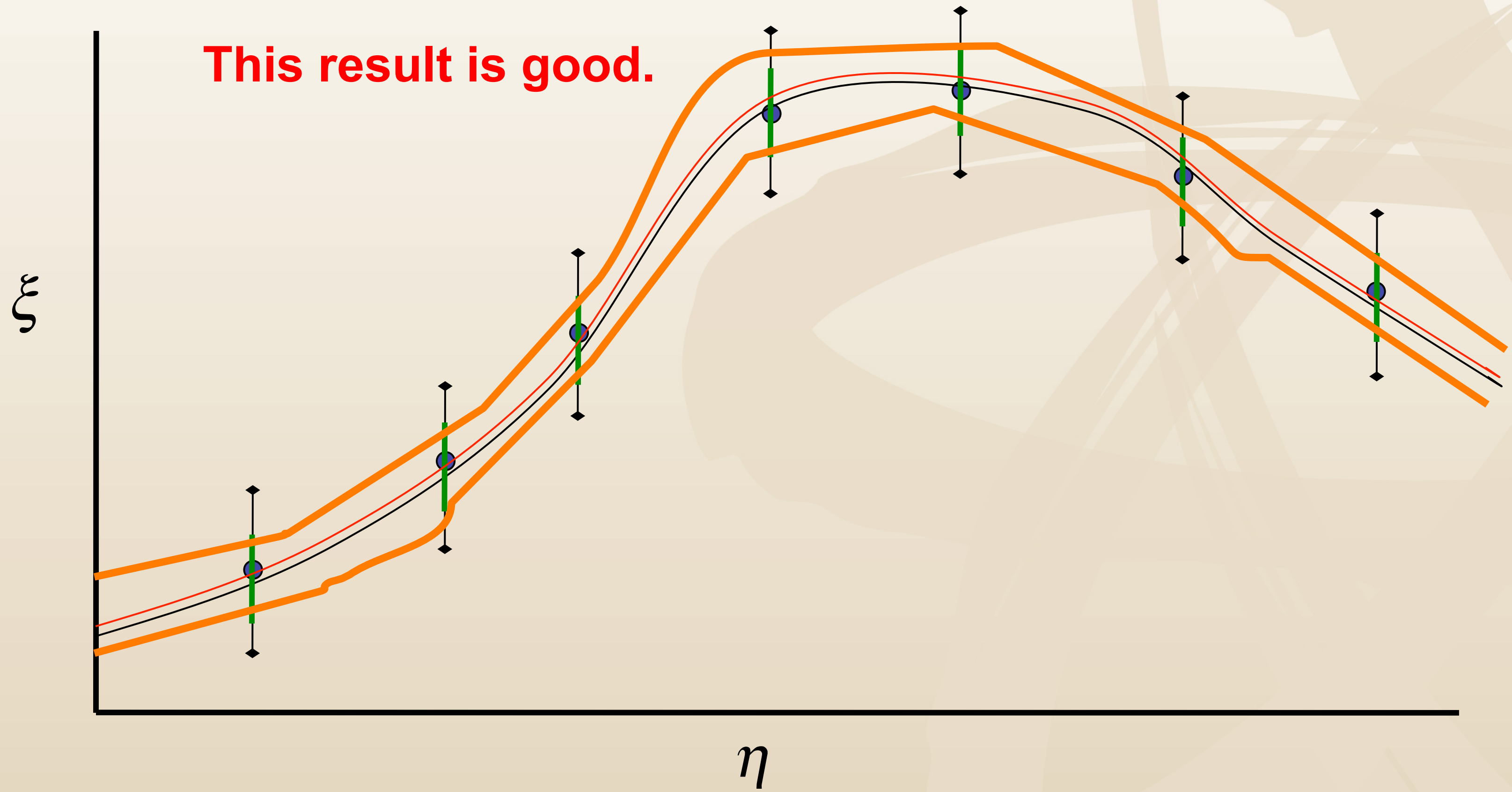
The experimental "error" has two components (observation & variability).



The model also can vary due to either stochastic or model coefficient uncertainty.

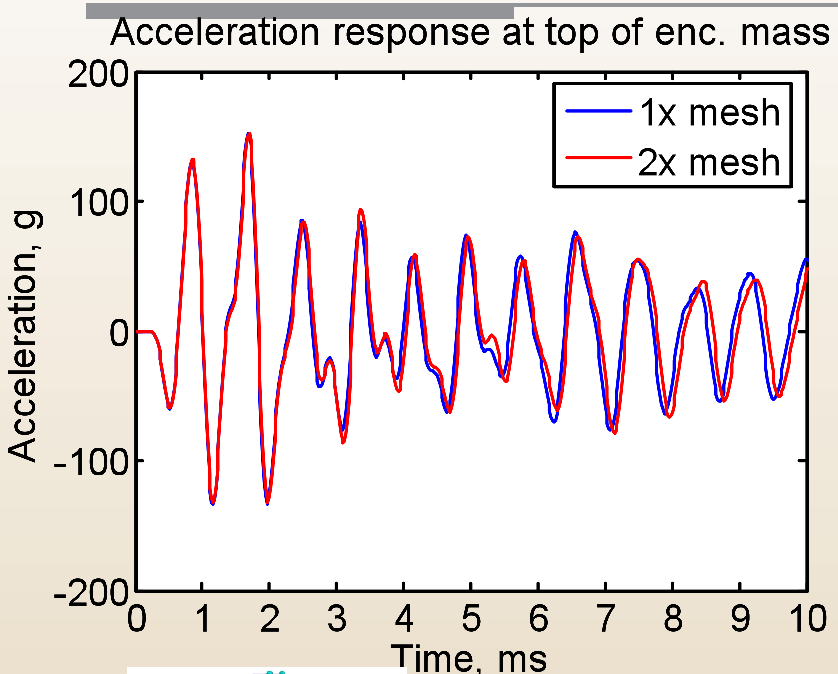


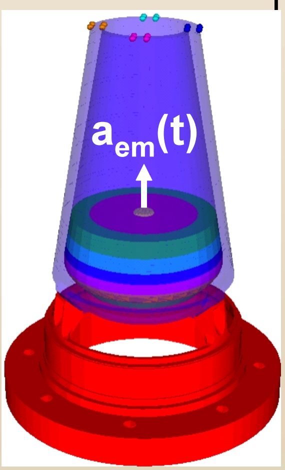
The model also can vary due to either stochastic or model coefficient uncertainty.



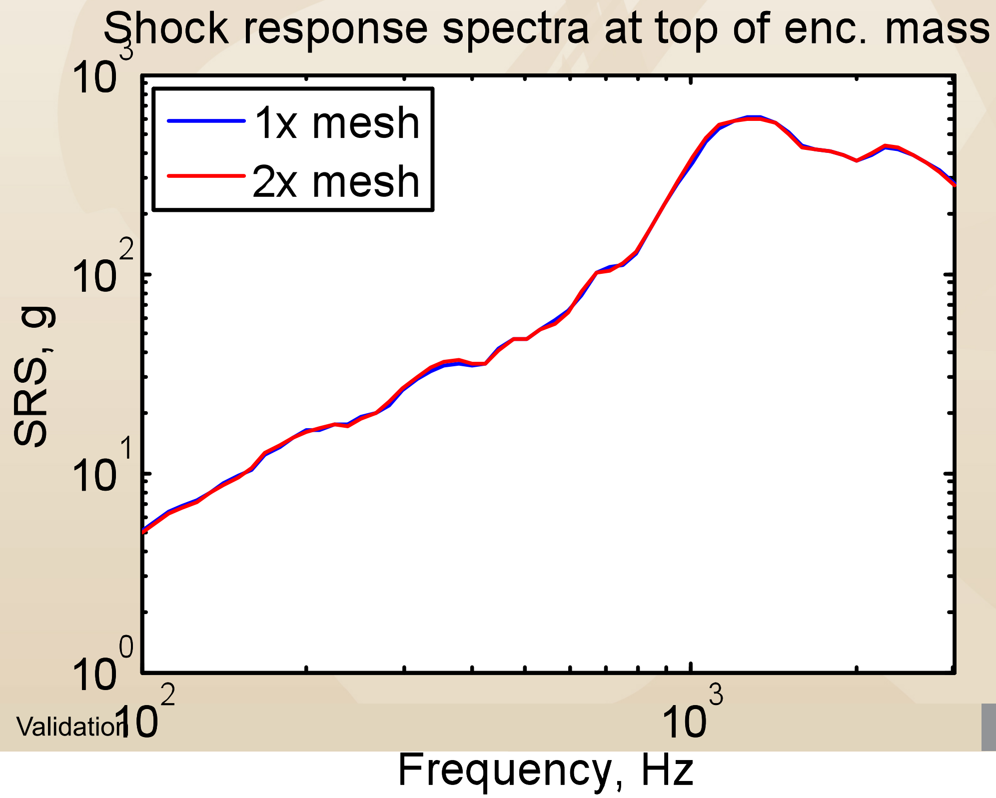
It's Common to Explore Sensitivity to Mesh Density

Slide from Marty Pilch's PCMM overview talk



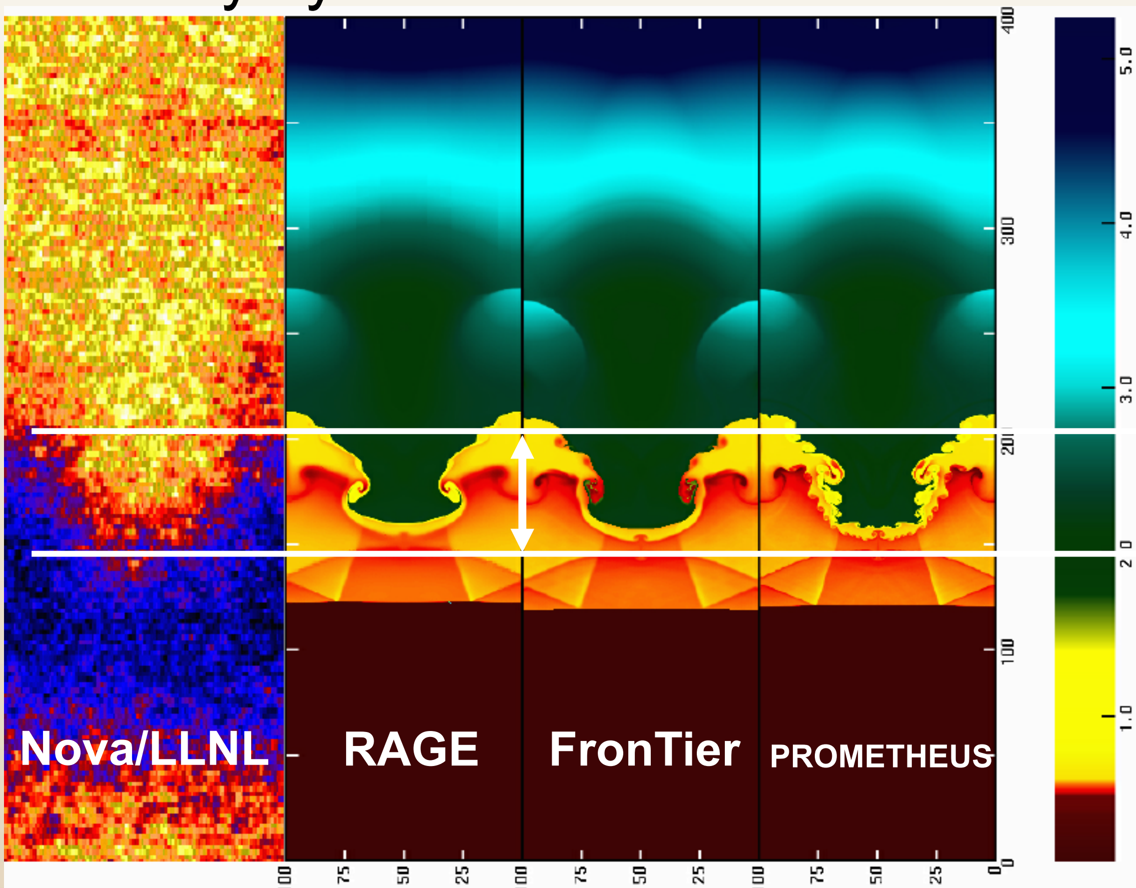


Max. relative error between SRS: +/- 5%



My starting point for looking at experimental measurement in validation.

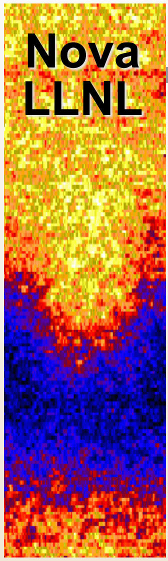
- Study by Holmes et al.* on R-M instability growth:

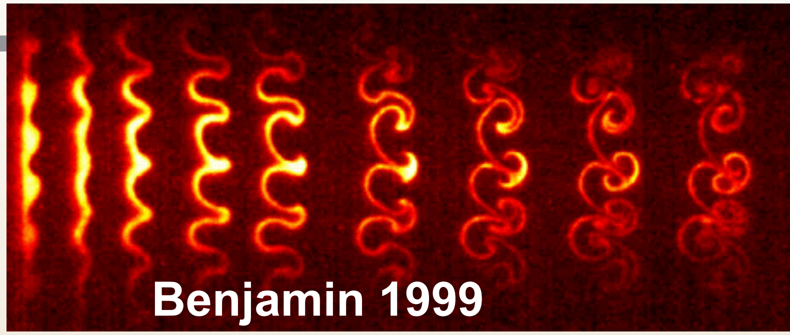


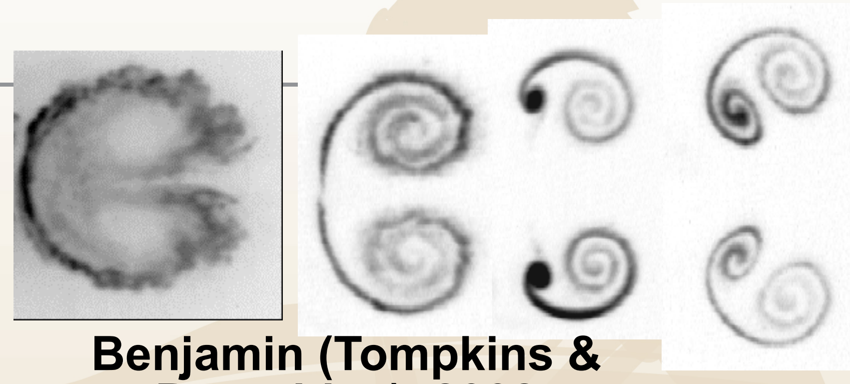
- ✴ Comparison of NOVA laser experiments with three different codes
- ✴ The integral scale seems to compare well— but how do the details compare? Can't tell from the data!

*Holmes et al., Richtmyer-Meshkov instability growth: experiment, simulation and theory, *J. Fluid Mech.* , 389 , pp. 55–79, 1999

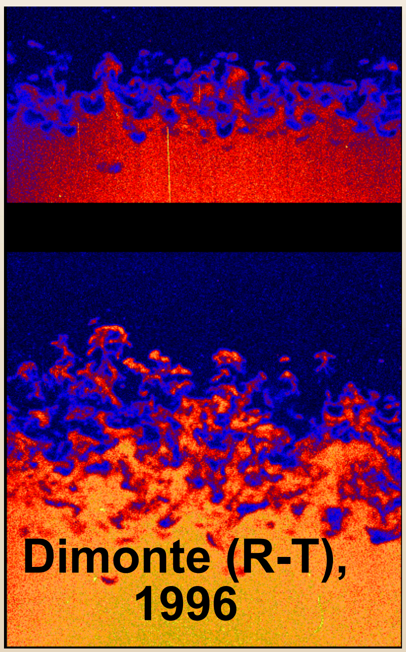
Importance of data quality in experiments

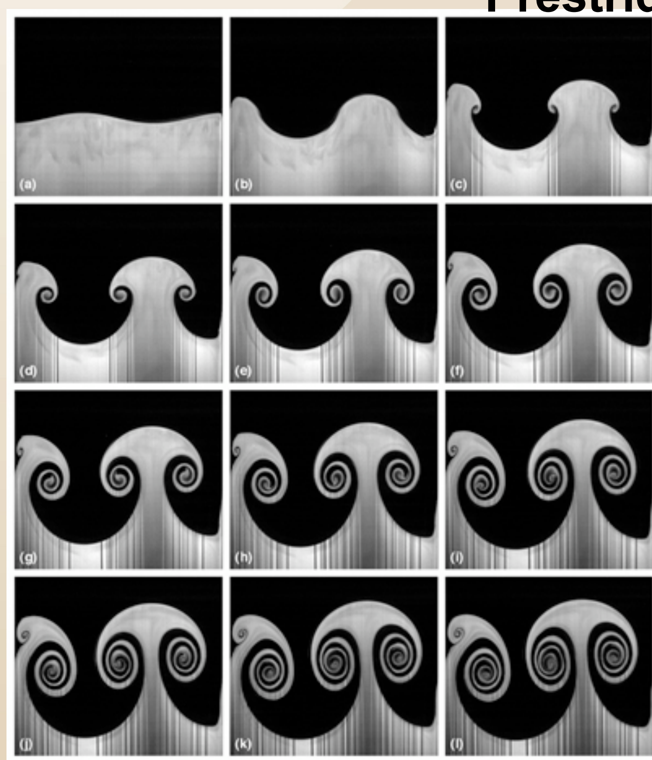


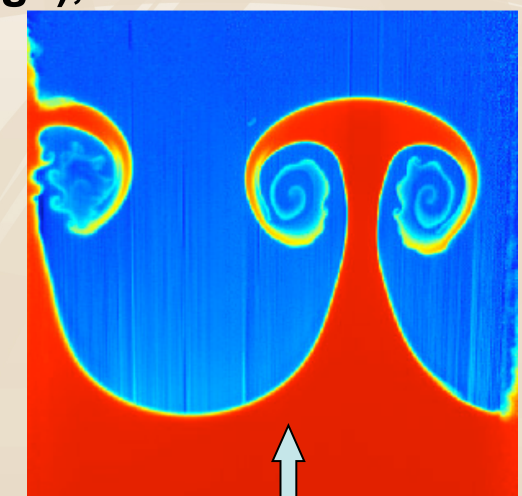




Benjamin (Tompkins & Prestridge), 2002

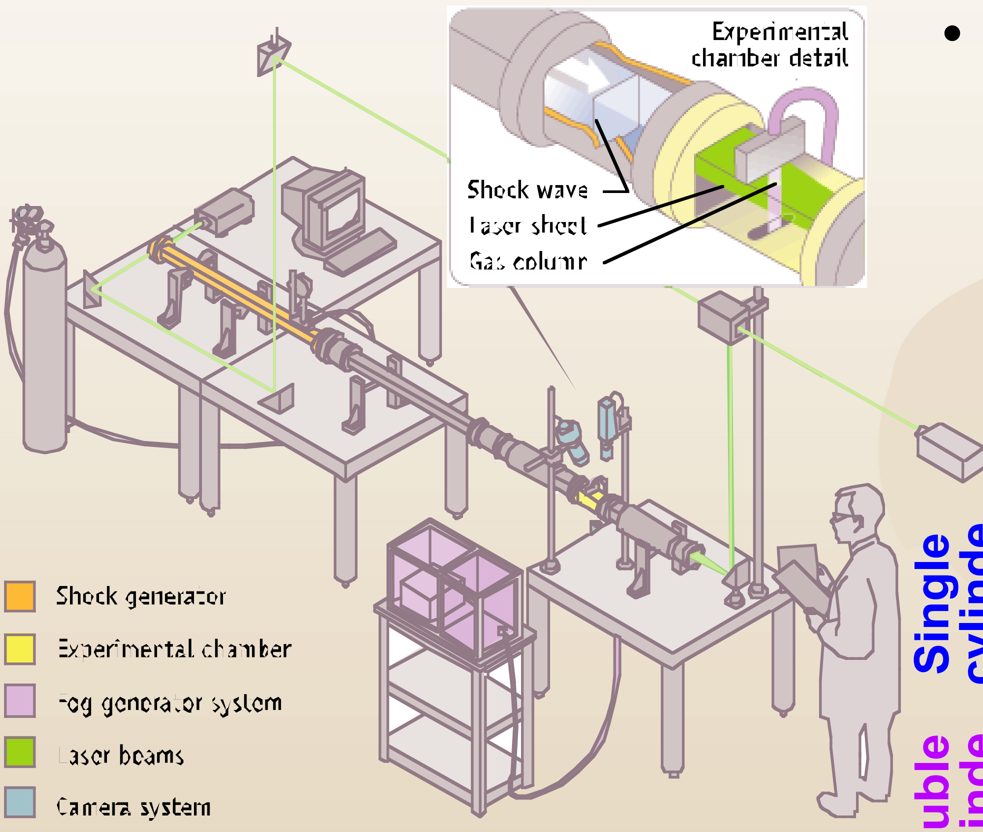


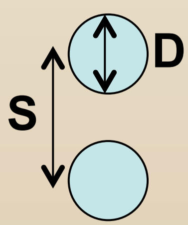




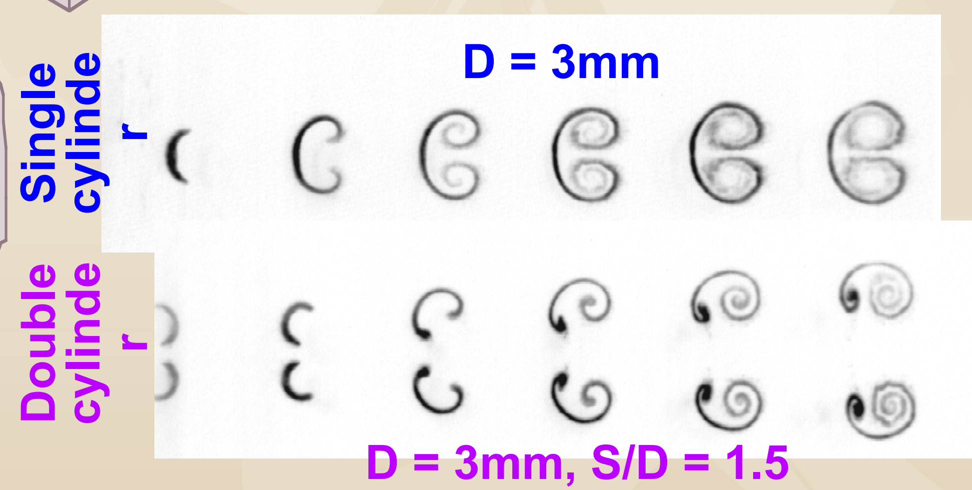
Jacobs, 2005

Shock Tube Experiments at LANL display a number of details to understand.



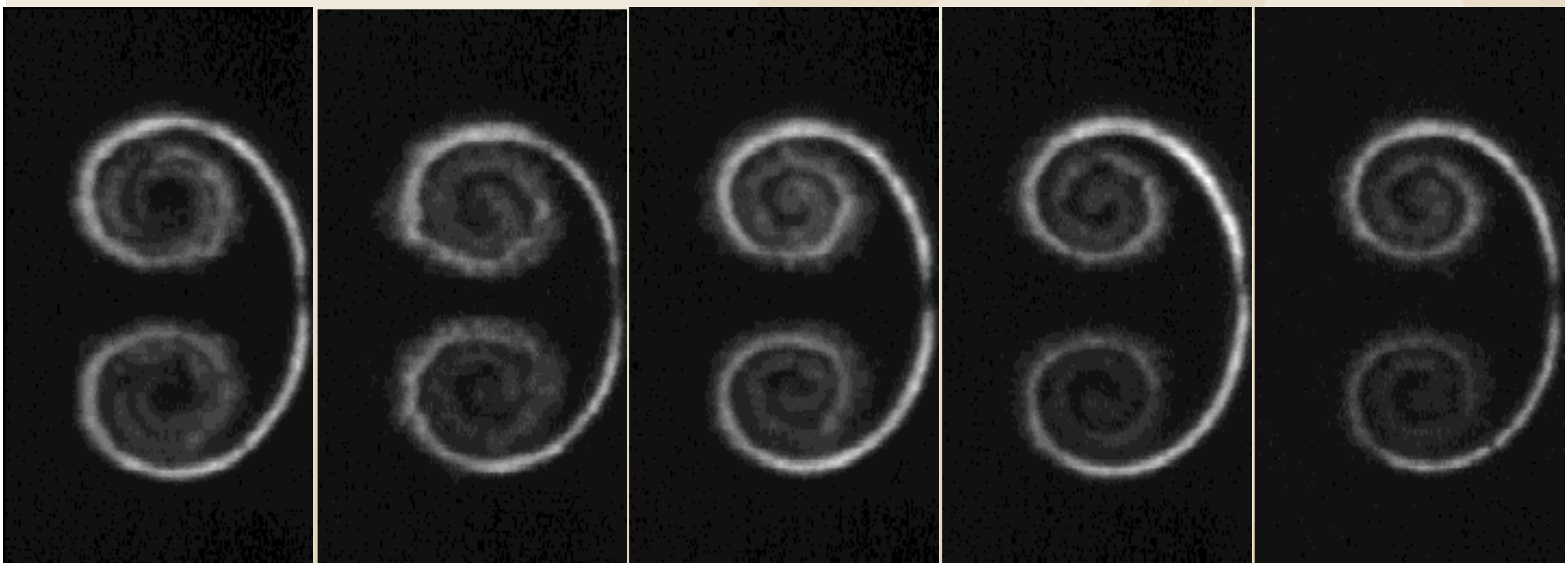


- Recent experiments have used gas cylinders as the target
  - **Experiments conducted by Benjamin, Prestridge, Rightley, Vorobieff (UNM), and Zoldi.**



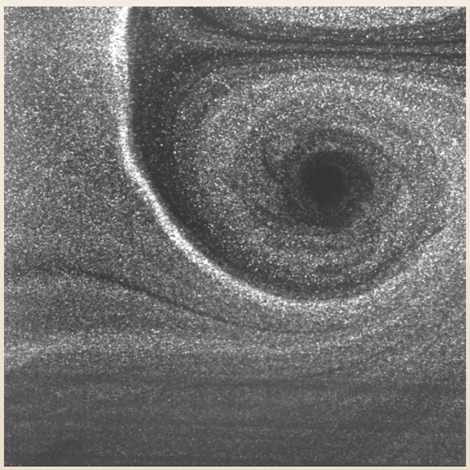
Shock Tube Experiments: Gas Cylinders

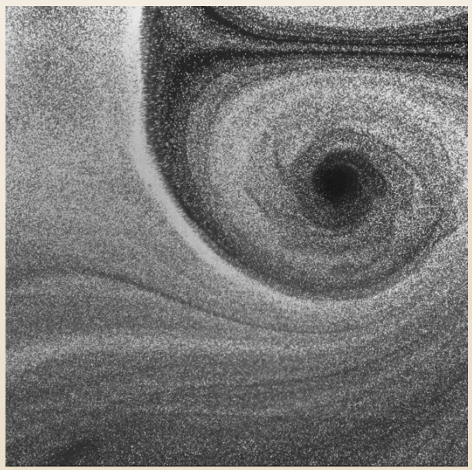
- Experiments are quite repeatable: ensemble statistics available using ~15 shots (variability can be described)
- The pixel size is relevant as is the exposure time for the image.
- We're next going to talk about initial conditions.

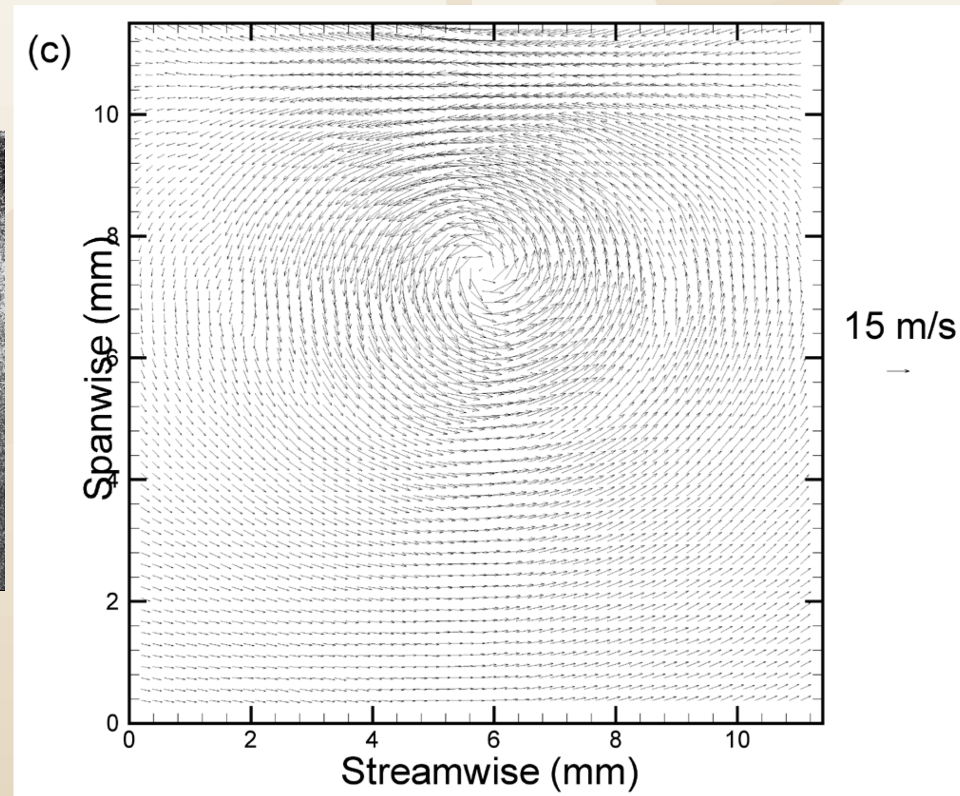


PIV data and analysis should provide a window into velocity & vorticity.

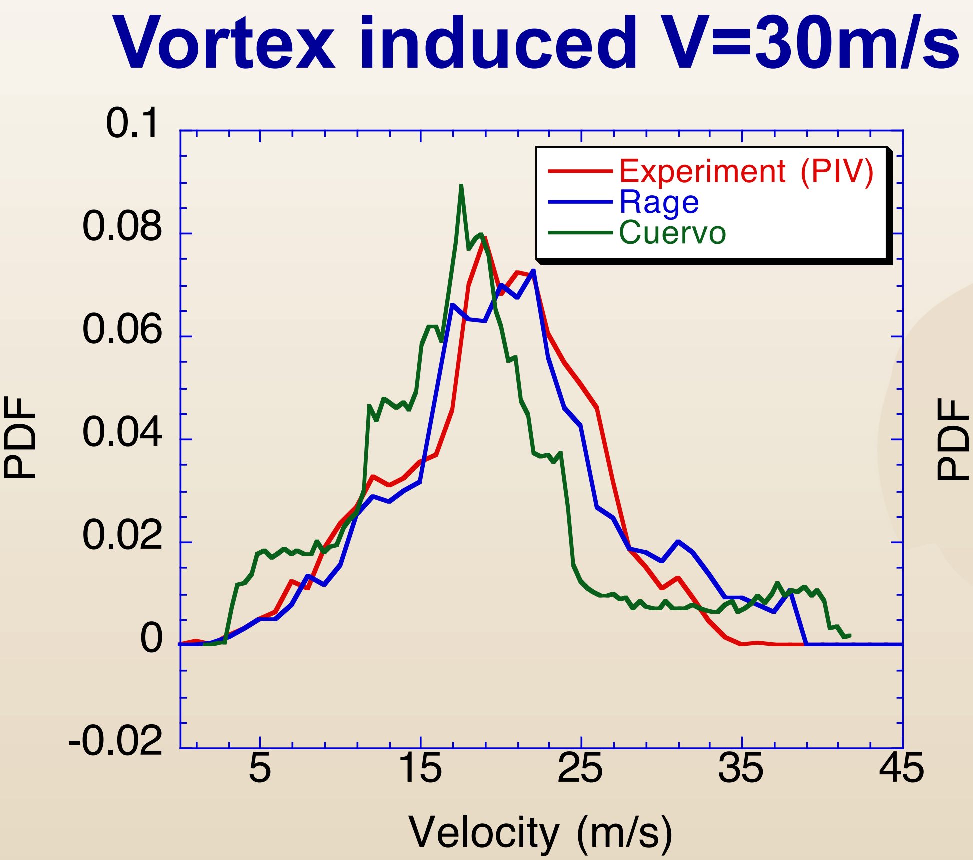
Analysis: two-frame cross-correlation (error from this)

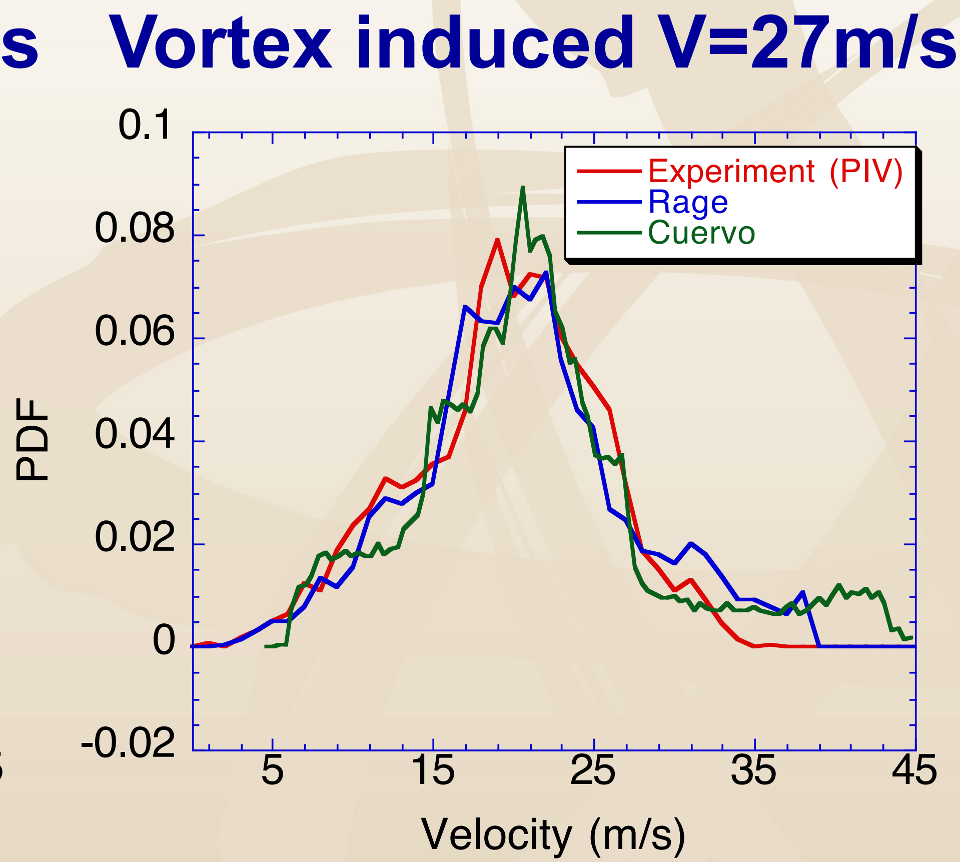






Comparison with PIV (we didn't include all the error analysis too!).

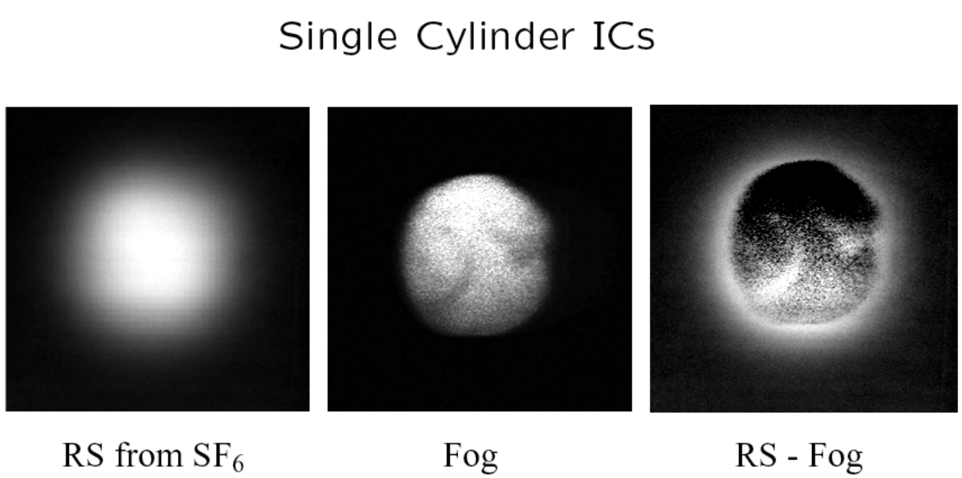


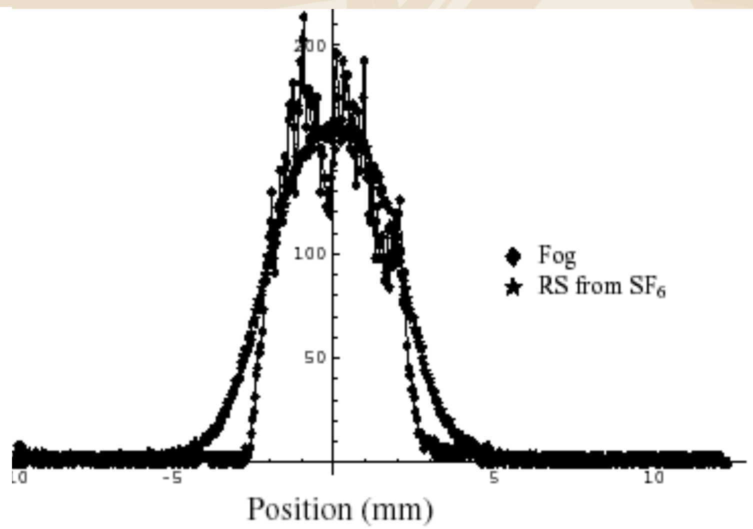


Simulations use compressible Navier-Stokes with Δx=10μm

Improved characterization of initial conditions provide important feedback.

Use Rayleigh scattering to see if the fog and SF_6 diffuse from each other. They do!



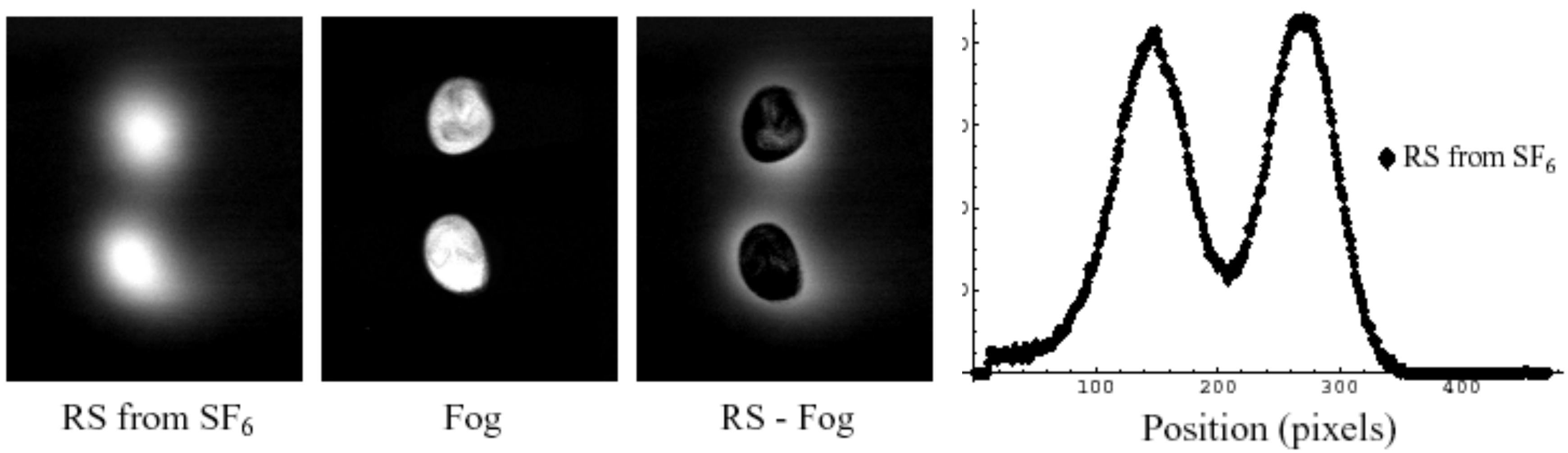


About 30% of the SF_6 is now outside the fog.

Improved Characterization of Initial Conditions

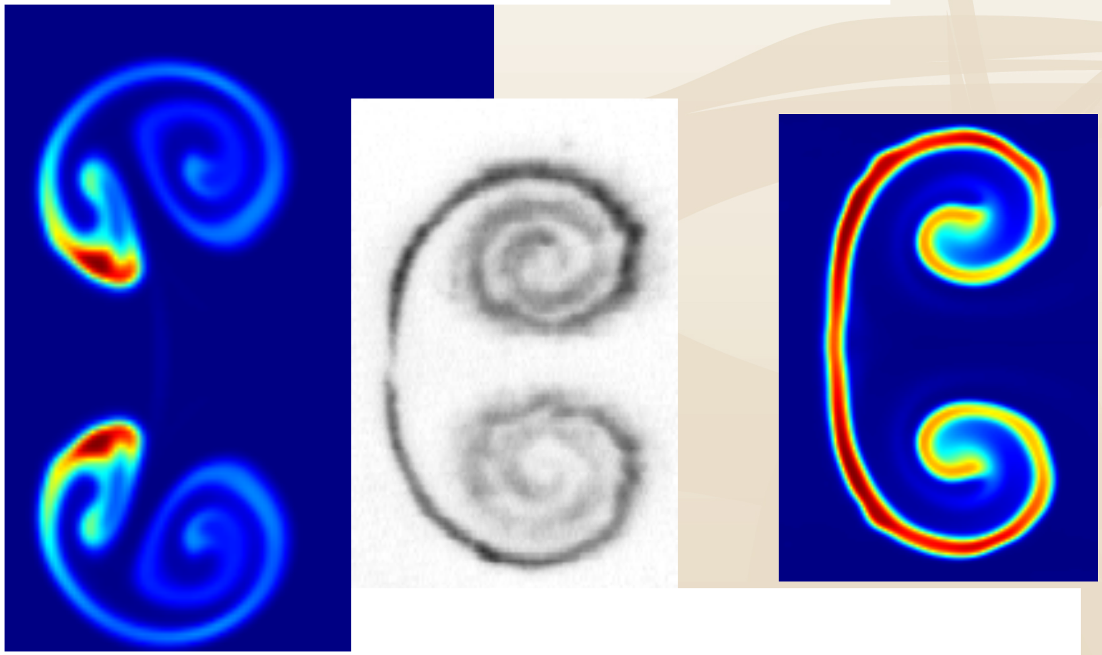
Use Rayleigh scattering to see if the fog and SF_6 diffuse from each other. For smaller cylinders a greater difference between the fog and SF_6.

Double Cylinder ICs: $S/D = 2.0$



The better initial conditions improve the quality of the simulations greatly.

S/D=1.2



Old IC Experiment New IC

Turbulent Mixing?

- Flowfield characterization -Is it turbulent?
  - ***It is transitioning to turbulent mixing***
  - **Single cylinder (5 mm)**

$$\mathrm{Re} = \Gamma/\nu \approx 70{,}000$$

$$\mathrm{Re}_T = \langle u \rangle \lambda_T / \nu \approx 800 - 2500$$

$$\lambda_T = \left(\langle u \rangle / \langle \partial u / \partial x \rangle \right) \approx 2mm - 5mm$$

Certainly not fully developed or isotropic!

  - **Double cylinder(3mm,S/D=1.5)**

$$\mathrm{Re} = \Gamma/\nu \approx 30{,}000$$

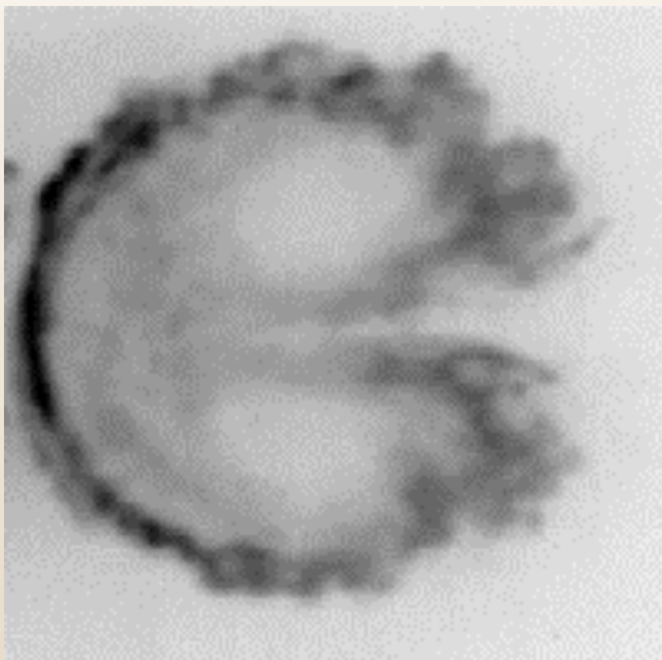
$$\mathrm{Re}_T = \langle u \rangle \lambda_T / \nu \approx 600 - 1500$$

$$\lambda_T = \left(\langle u \rangle / \langle \partial u / \partial x \rangle \right) \approx 2mm - 4mm$$

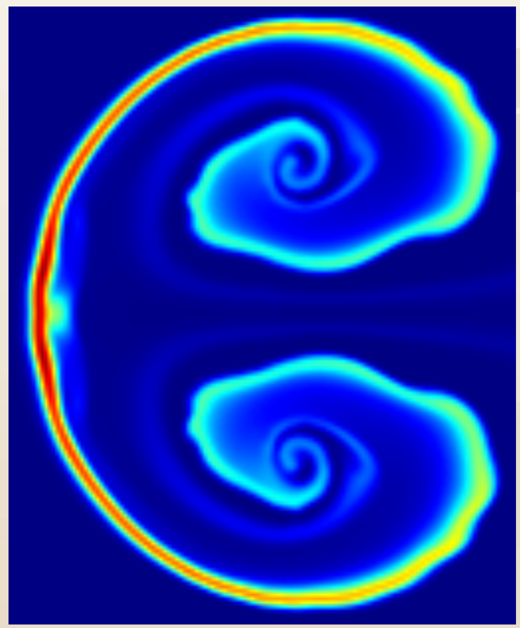
$$\eta = \begin{cases} 0.01mm \\ 1\mu m \end{cases}$$

Single Cylinder with New ICs

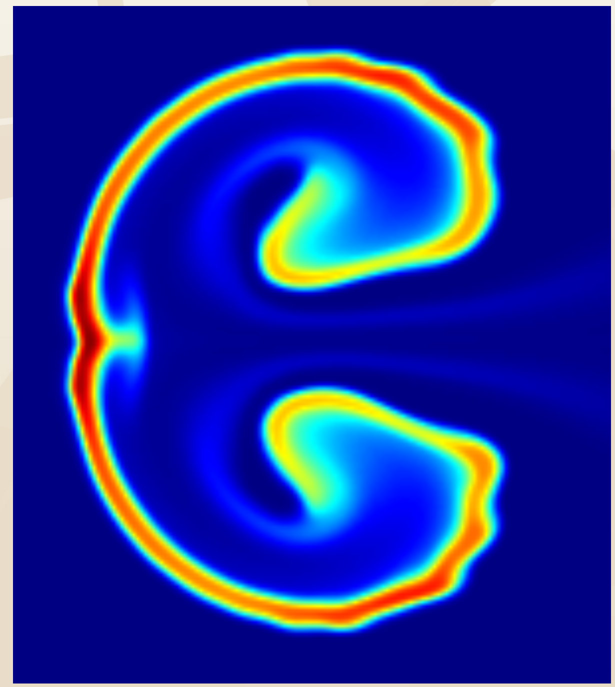
Experiment



New Method



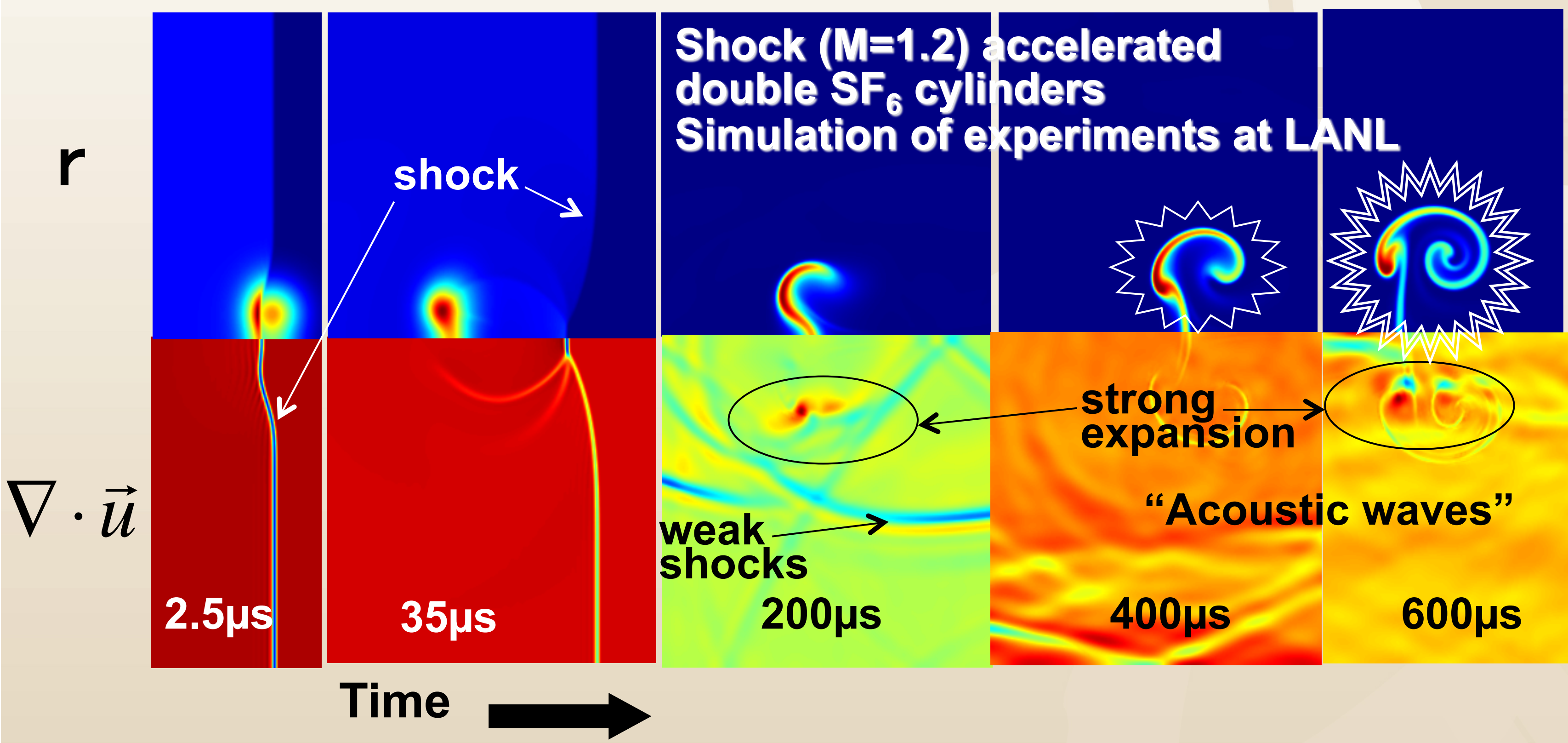
Old Method



Compressible Navier-Stokes Δx=10µm

Using Simulations for Qualitative Information?

- Compressible flow leading to mixing

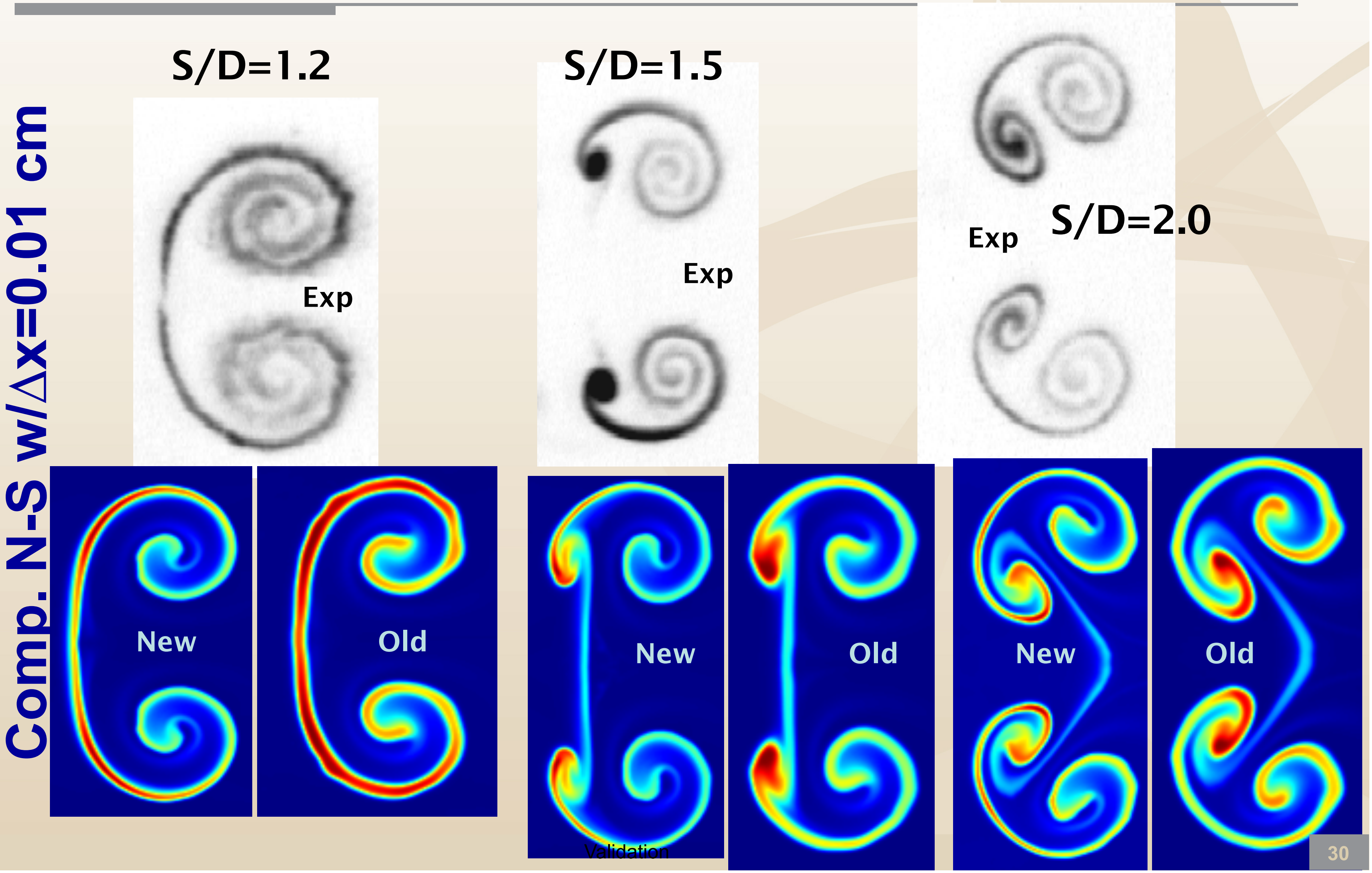


Shocks are "easy,"

Mixing is "hard"

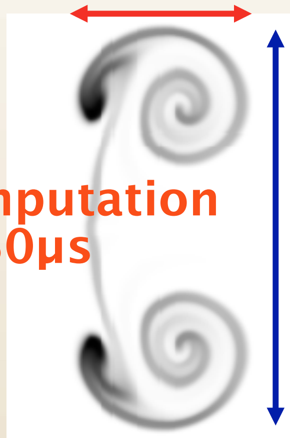
– Its driven by the baroclinic source $\nabla\rho \times \nabla p \Rightarrow \nabla \times \vec{u}$

Double Cylinder with New ICs



Example results S/D=1.5 at 750µs show that the agrement needs work.

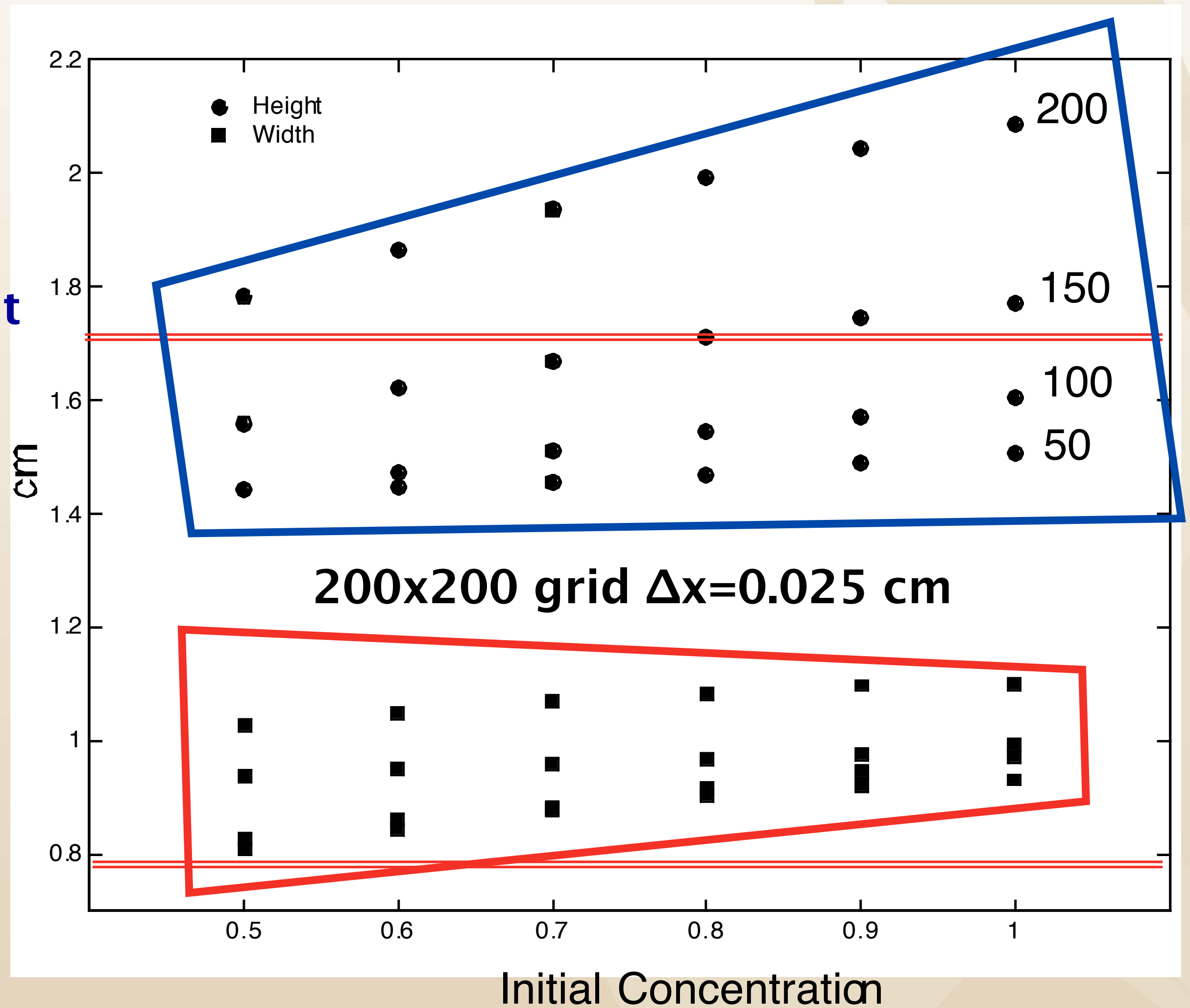
width



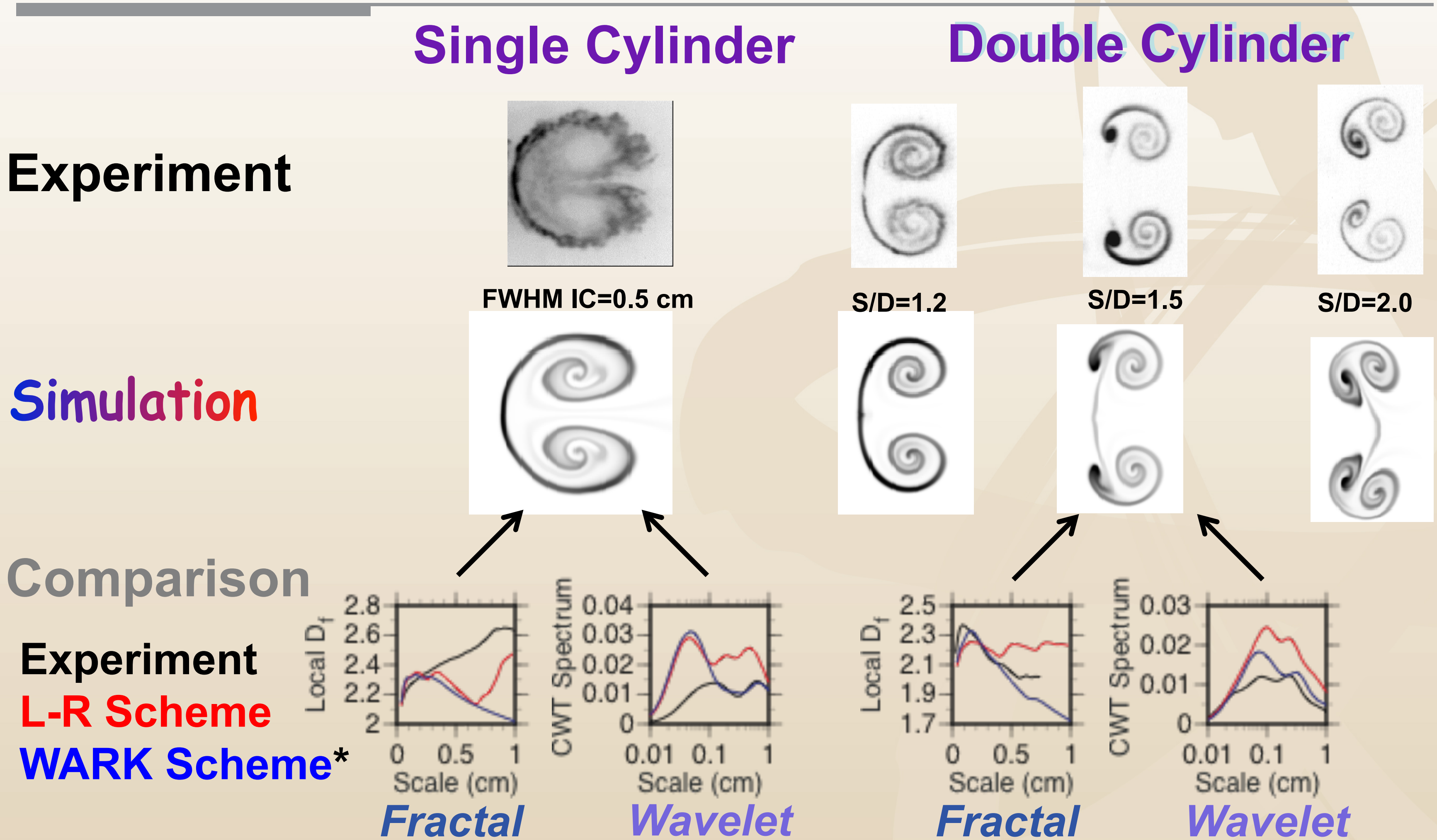
height

Computation
@750µs

Experiment
@750µs

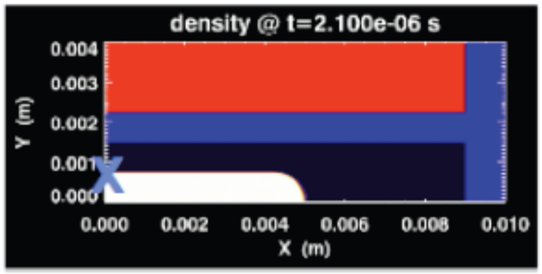


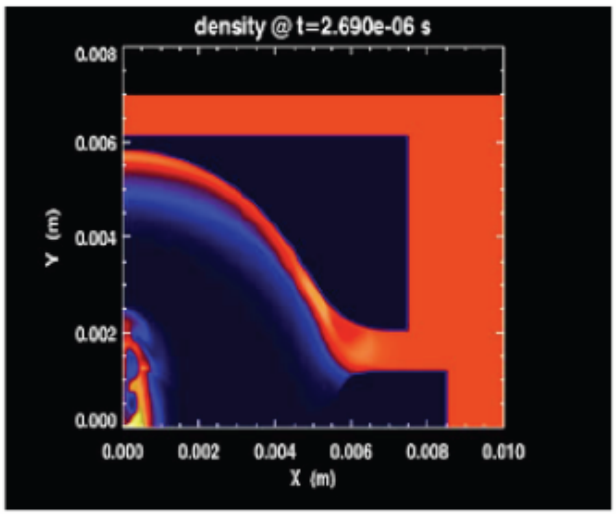
This shows a multiscale comparison of the shock tube data with calculations.

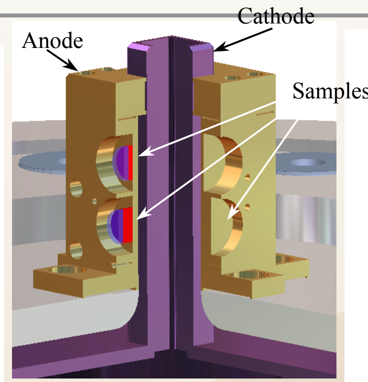


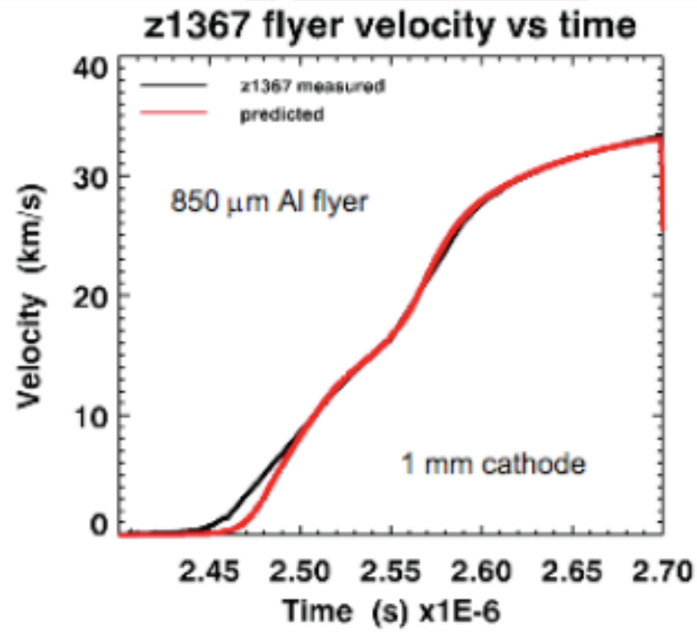
*Rider et al., “An Adaptive Time Integration Algorithm for Hyperbolic Conservation Laws”, Proc. 9th Int’l Conf. Hyperbolic Problems, to appear, 2002.

Simulation of a magnetic flyer has been a strength of ALEGRA.



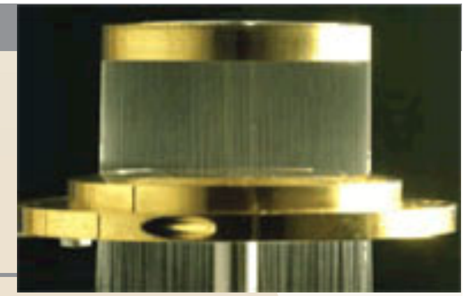


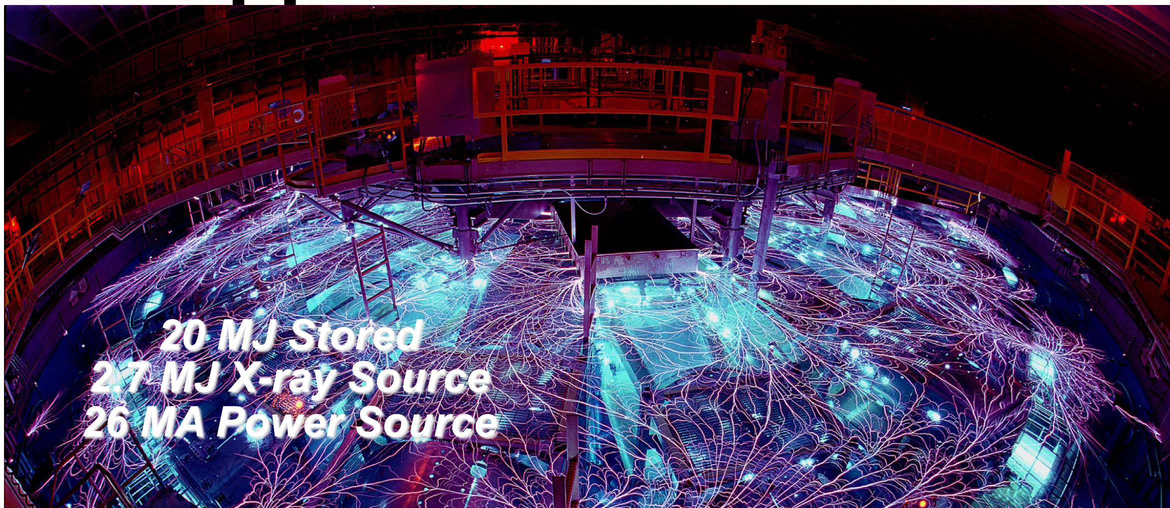


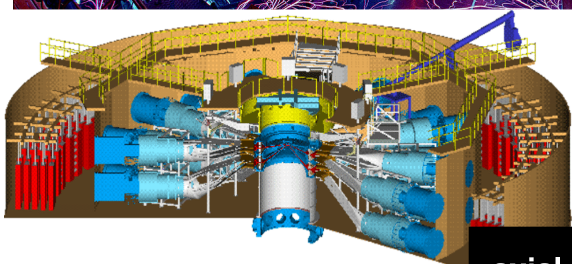


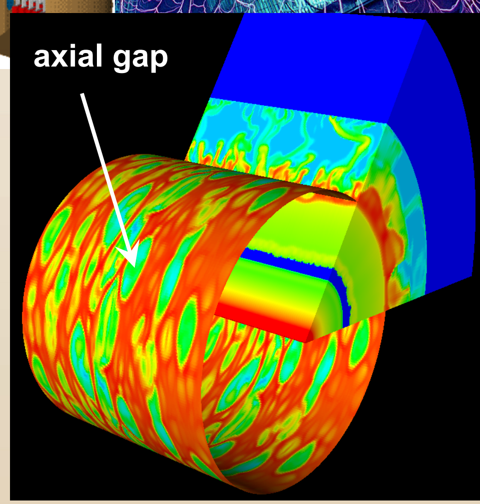
No error bars, later work has estimated the numerical error at 2%
The largest error is the drive.
The measurement error is about 1%

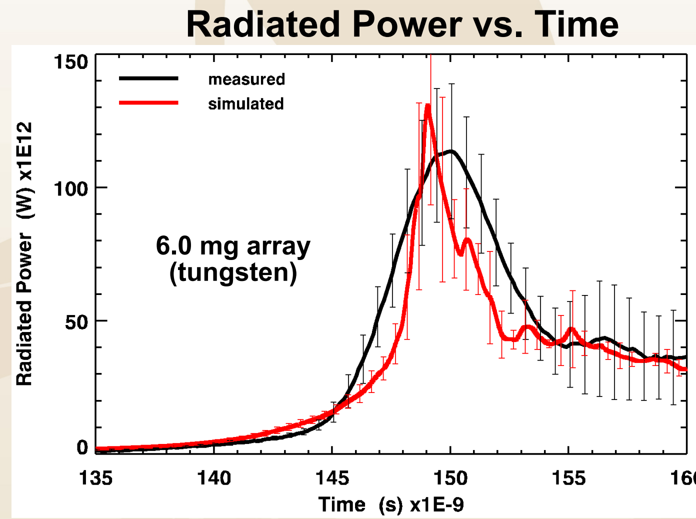
Applications: Z-Pinch*

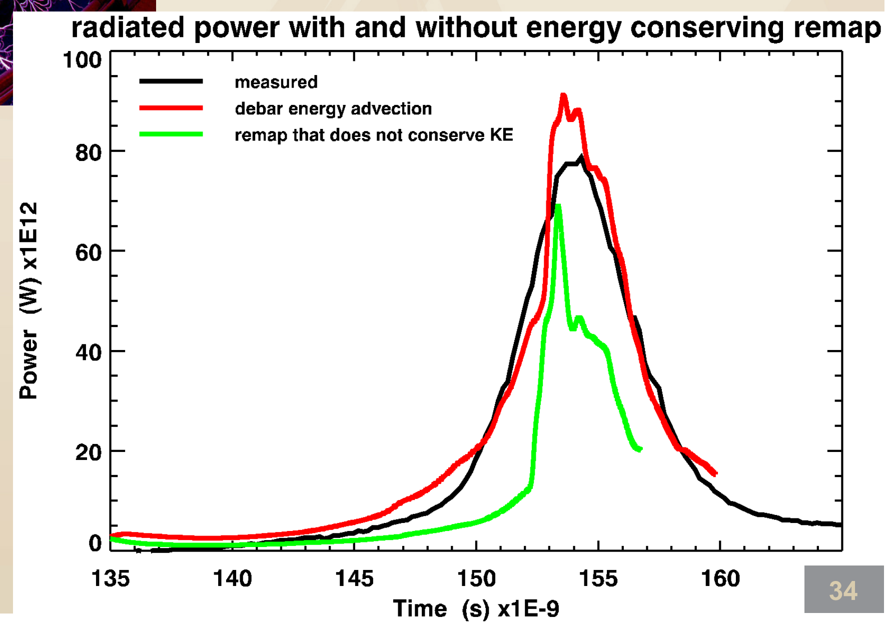








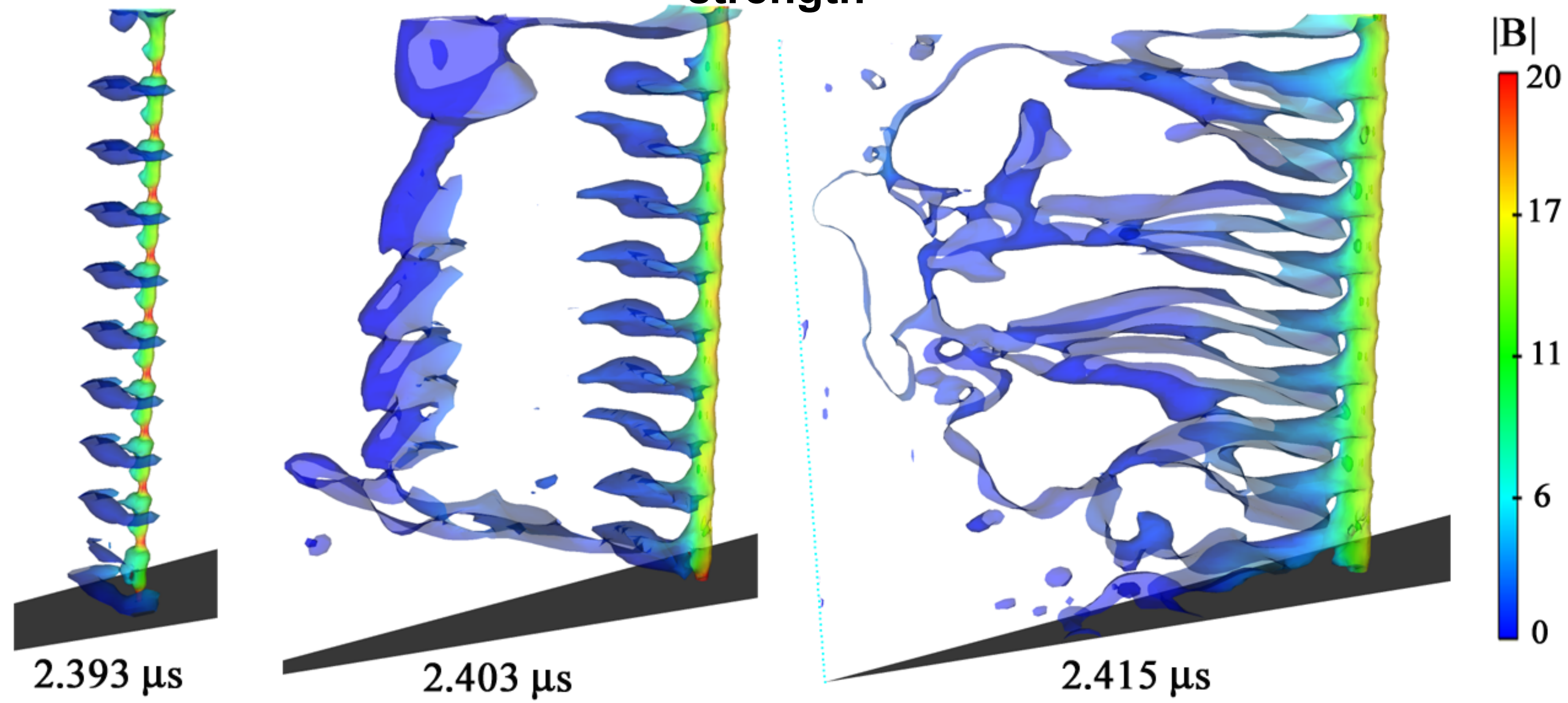




* work by Lemke, et. al

Sinusoidal Core Perturbation

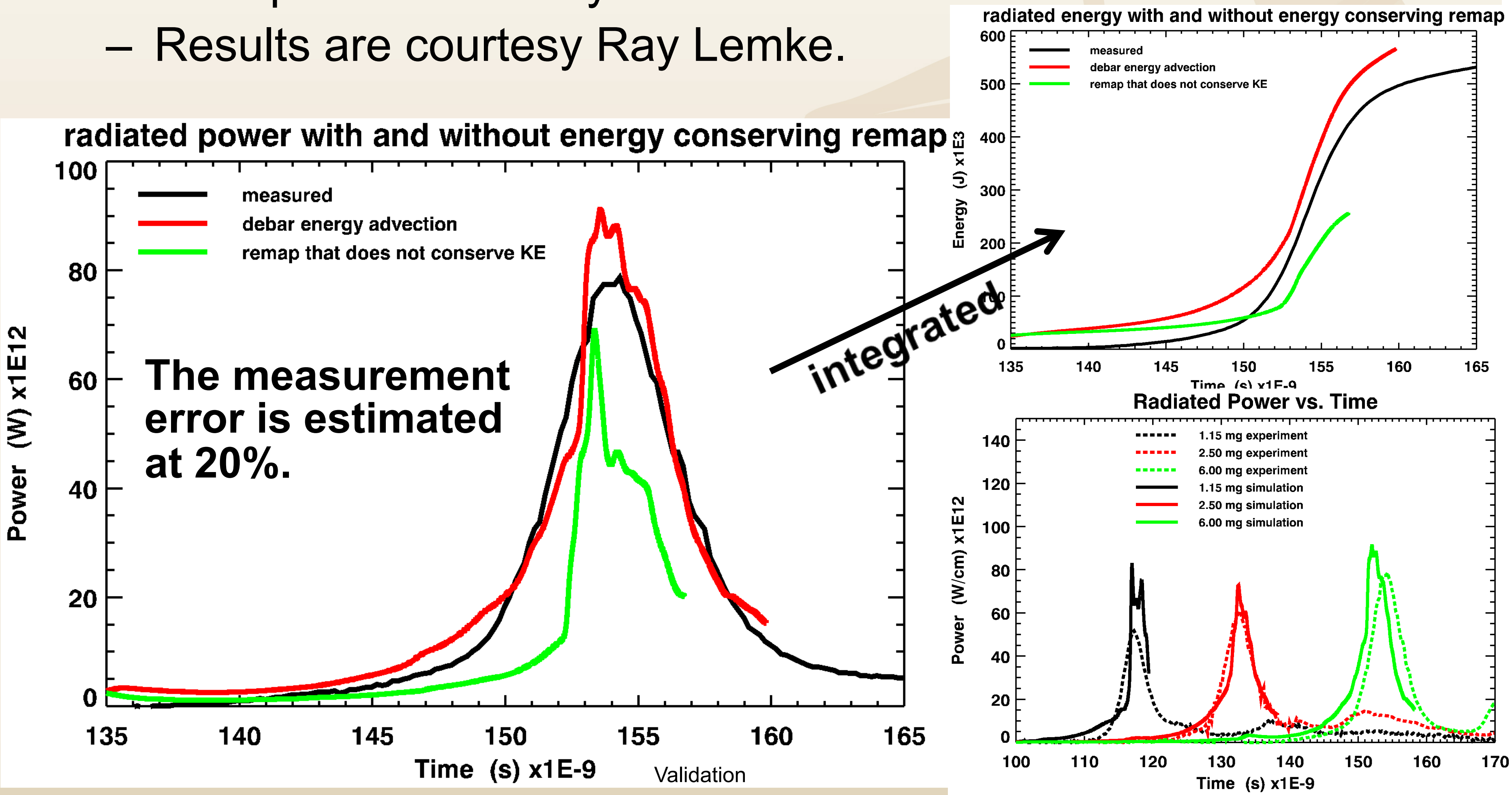
Volume fraction isosurface with magnetic field strength



- This shows the details that explain the dynamics of the wire array implosion
- It is basically a magnetic Rayleigh-Taylor stagnating on center

Algorithm Impact: 3D Z-Pinch Implosion

- This shows the impact of using a better energy remap. The radiated power is the key metric.
  - Results are courtesy Ray Lemke.

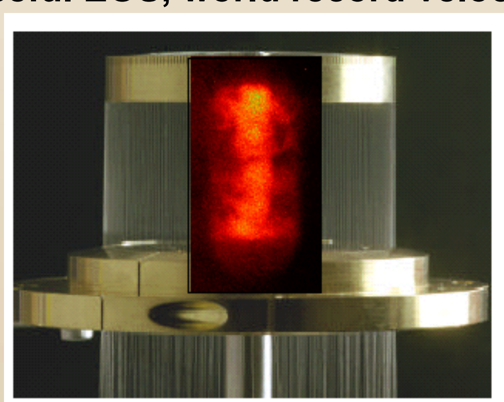


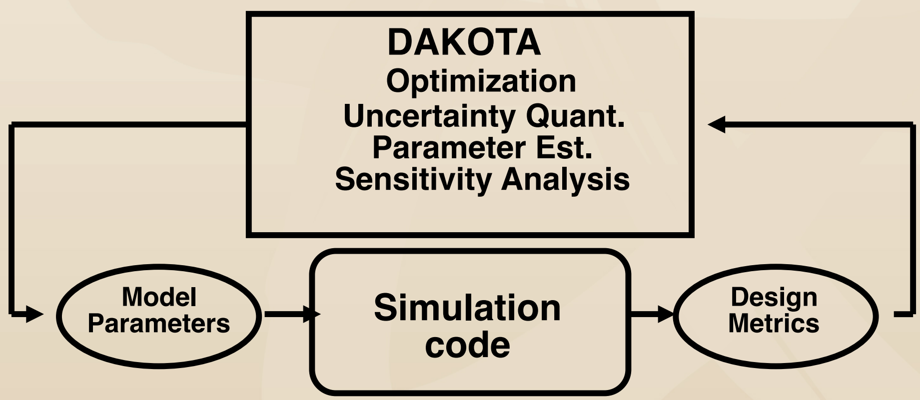
Validation

Dakota contains many of the tools necessary to advance.

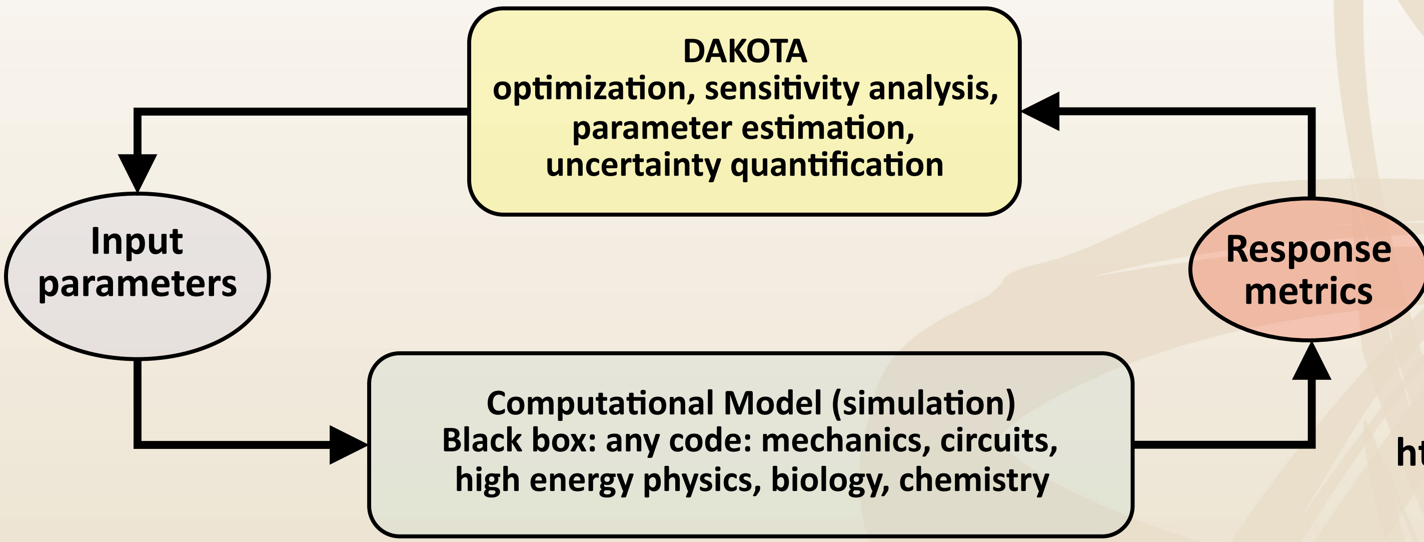
- In a sense Dakota is a toolbox of techniques that can be used to examine a code's results.
- Included are optimization, sensitivity and uncertainty quantification (see application, next!).
- An important aspect is the use of "design of experiment" or sampling techniques for integrating high-dimensional space which is necessary for UQ etc.

Z pinch flyer plate
(useful EOS, world record velocities)





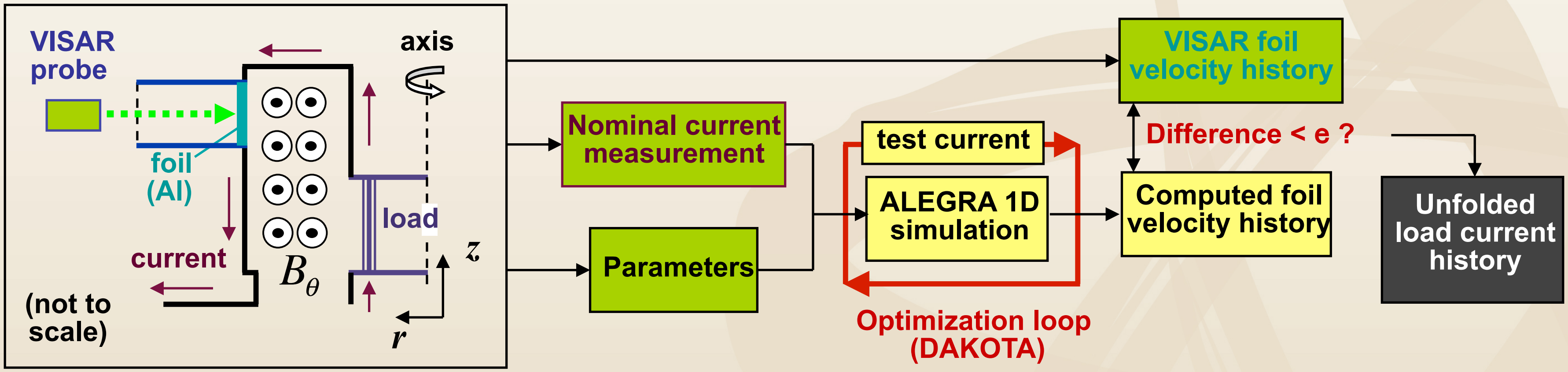
DAKOTA could be used for sensitivity and uncertainty analysis among other things.



http://dakota.sandia.gov/

- DAKOTA can automate typical "parameter variation" studies with a generic interface to simulation software and advanced methods.
- UQ methods in DAKOTA include:
  - Sampling (LHS, quasi-MC, classical experimental designs, OAs, VBD)
  - Reliability methods (FORM, SORM, AMV+, etc.)
  - Dempster-Shafer Evidence Theory
  - Stochastic expansion methods: Polynomial chaos, stochastic collocation
  - Epistemic-aleatory nested approaches

Load-current unfold procedure: uncertainty quantification needed



New diagnostic in Z-facility experiments unfolds load current from measured foil velocities using ALEGRA simulations with optimization:

UQ efforts for load-current unfold procedure: uncertainties & sensitivities evaluated

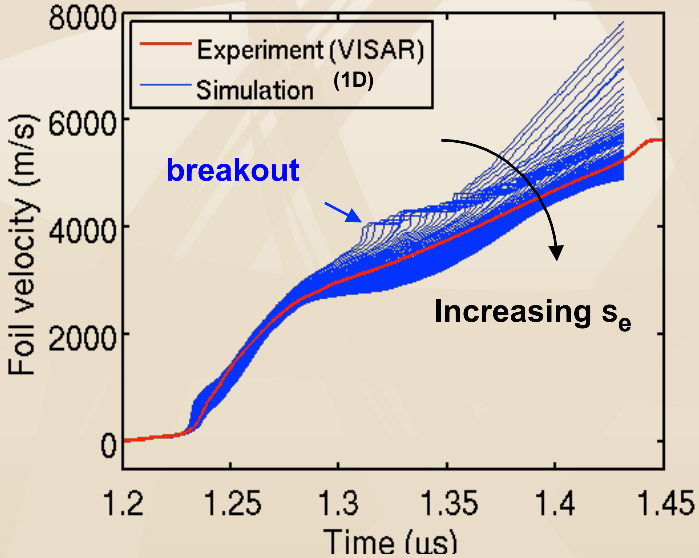
Numerical error in computed velocity histories:

Simulation parameter	Associated fractional uncertainty in velocity
Grid size	0.41%
Timestep size	0.12%
Solver tolerance	0.01%

- Obtained from solution verification with Richardson extrapolation for 1D forward problem.
- Total numerical error for forward problem is 0.43% in velocity; propagation to unfolded current is needed.

Sensitivity in computed velocity histories to s_e:

- We have established that the velocity history is very highly sensitive to tabulated electrical conductivities s_e.
- More significantly, an inflection point in v(t) indicates magnetic field breakout and varies linearly under ±10% variation in s_e.
- Currently investigating: can variation in breakout time t_b relative to experiment be used to estimate error in s_e?



Error estimates for current unfold:

- Strongest contributors to uncertainty in current: (1) uncertainty in s_e, (2) discretization error.
- With error bounds on s_e and convergence rate of simulation established, standard UQ methods can be used to compute current-unfold uncertainty.

Predictive Capability Maturity Model (PCMM) is being used to gauge codes.

- PCMM is a framework for examining the status and progress in simulation.
- The process is to "grade" codes and the simulations they produce in several key areas.
- This includes the users of the code and their expectations and future needs.
- In a sense its loosely modeled on the CMM in software engineering.

Example of PCMM in Action

Category/ Level	Level 0 (no assessment)	Level 1 (informal assessment)	Level 2 (some assessment)	Level 3 (formal assessment)
Geometric fidelity			assessed	required
Physics fidelity			assessed	required
Code Verification		assessed	required	
Solution Verification	assessed	required		
Model Validation		assessed	required	
UQ + Sensitivity		assessed	required	

Verification

Who?: Code developers & mathematicians, algorithm engineers
What?: compare code solutions with analytical solutions.
How?: Mesh refinements studies, computing errors and error estimates.
Why?: To make sure that a model in a code is implemented correctly.
Confused with?: Software quality assurance (SQA), benchmarking, patch test, code comparison
What is hard?: The analytical structure of solutions is not always known. The *verification* studies are quite tedious.

Validation

Who?: Code developers, users, and modelers.
What?: Compare code solution results with experimental data or observations.
How?: Comparing the data along with its intrinsic errors with the code model result.
Why?: To determine whether to model in the code is a high fidelity model of reality.
Confused with?: *calibration*, application modeling, uncertainty quantification
What is hard?: Models in codes can have errors ranging from coding to conceptual in nature. Data and observations are often poor in quality and control with undefined or poorly characterized errors.

Benchmarking

Who?: code developers and users
What?: define standard problems that are similar to application problems, but easier to set up or examine.
How?: Often one code will be used to define a "standard," justified or not.
Why?: It is easier than verification and validation. Success with *benchmarking* often grants to developers and users a "warm fuzzy feeling" about a code.
Confused with?: *verification, validation*
What is hard?: It is a fairly easy activity although doing a credible job of developing a "good" benchmark is a challenge.

Calibration

Who?: Code users, application modelers.
What?: Pick parameters to give good matches to data and observations.
How?: Trial and error, optimization of results. Also known as a "knob".
Why?: It is a pragmatic activity used when the model cannot produce acceptable results in a scientifically justifiable manner.
Confused with?: *validation*, science
What is hard?: Producing quantifiably "good" results with an inherently flawed physical model is not easy.

UQ: Uncertainty Quantification

Uncertainty quantification is the study of the size and causes for simulation uncertainty. Uncertainty comes in two main flavors: aleatory meaning intrinsic, and epistemic meaning from lack of knowledge. Different techniques exist for using a model or data in such a way to produce evidence of the magnitude and nature of uncertainties.

QMU

Quantified margins and uncertainties is a “framework” for high consequence decision making developed by the DOE Labs to help with stockpile stewardship. QMU uses V&V + UQ to help provide an evidence based approach to its advice to the country.

PCMM

The predictive capability maturity model is a system for identifying the quality of and the needs for simulation and modeling in the analysis of systems. It includes all the aspects of V&V + UQ as well as a “grading” system for deciding what is “good enough”.